Administrator's Guide

iPlanet Web Server, Enterprise Edition

Version 6.0 SP1

816-1379-01
August 2001

THIS DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

CETTE DOCUMENTATION EST FOURNIE “EN L'ÉTAT”, ET TOUTES CONDITIONS EXPRESSES OU IMPLICITES, TOUTES REPRÉSENTATIONS ET TOUTES GARANTIES, Y COMPRIS TOUTE

Contents

Chapter 16 Content Management 345

Chapter 17 Applying Configuration Styles 361

About This Guide

This guide describes how to configure and administer iPlanet™ Web Server, Enterprise Edition 6.0. It is intended for information technology administrators in the corporate enterprise who want to extend client-server applications to a broader audience through the World Wide Web.

This preface includes the following sections:

- What’s In This Guide?
- How This Guide Is Organized
- Conventions Used In This Guide
- Using the iPlanet Web Server Documentation
- Further Reading
- Contacting Technical Support

What’s In This Guide?

This guide explains how to configure and administer the iPlanet Web Server. After configuring your server, use this guide to help maintain your server.

After you install the server, this guide is available in HTML format at `manual/https/ag` in your server root directory. By default, the server root directory is `C:\iPlanet\Servers\` or `/usr/iplanet/servers`.

How This Guide Is Organized

This guide is divided into five parts, plus a glossary, and a comprehensive index. If you are new to iPlanet Web Server, Enterprise Edition 6.0, begin with Part I, “Server Basics” for an overview of the product. If you are already familiar with this version of iPlanet Web Server, skim the material in Part I, “Server Basics” before going on to Part II, “Using the Administration Server.”

Once you are familiar with the fundamentals of using the Administration Server, you can refer to Part III, “Configuring, Monitoring, and Performance Tuning,” which includes examples of how to configure and monitor your iPlanet Web Servers. Part IV, “Managing Virtual Servers and Services” provides information for using programs and configuration styles.

Finally, Appendixes addresses specific reference topics that describe the various topics, including: HyperText Transfer Protocol (HTTP), server configuration files, ACL files, internationalization issues, server extensions, and the iPlanet Web Server user interface reference, which you may want to review. Note that the user interface appendix is available in the online version only.

Part I: Server Basics

This part provides an overview of the iPlanet Web Server. The following chapters are included:

- Chapter 1, “Introduction to iPlanet Web Server” provides an overview of iPlanet Web Server.
- Chapter 2, “Administering iPlanet Web Servers” describes how to manage your iPlanet Web Servers with the Administration Server.

Part II: Using the Administration Server

This part provides conceptual and procedural details about using the Administration Server to administer your iPlanet Web Servers. The following chapters are included:

- Chapter 3, “Setting Administration Preferences” describes how to use the Administration Server Preferences and Global Settings forms to configure your iPlanet Web Servers.
- Chapter 4, “Managing Users and Groups” describes how to how to use the Administration Server Users and Groups forms to configure your iPlanet Web Servers.
- Chapter 5, “Securing Your Web Server” describes how to configure your iPlanet Web Server security. Note that before reading this chapter you should be familiar with the basic concepts of public-key cryptography and the SSL protocol. These concepts include encryption and decryption; keys; digital certificates and signatures; and SSL encryption, ciphers, and the major steps of the SSL handshake.
- Chapter 6, “Managing Server Clusters” describes the concept of clustering servers and explains how you can use them to share configurations among servers.

Part III: Configuring, Monitoring, and Performance Tuning

This part includes examples of how to use the Server Manager to configure and monitor your iPlanet Web Servers. The following chapters are included:

- Chapter 7, “Configuring Server Preferences” describes how to configure server preferences for your iPlanet Web Server.
- Chapter 8, “Controlling Access to Your Server” describes how to specify who can access parts of your server.
- Chapter 9, “Using Log Files” describes how to monitor your iPlanet Web Server using the Hypertext Transfer Protocol (HTTP), by recording and viewing log files, or by using the performance monitoring tools provided with your operating system.
- Chapter 10, “Monitoring Servers” describes how to monitor your iPlanet Web Server using SNMP (Simple Network Management Protocol).
- Chapter 11, “Tuning Your Server for Performance” refers you to the online document, *Performance Tuning and Sizing Guide* found in the following location: `http://docs.iplanet.com/docs/manuals/enterprise.html.`
- Chapter 12, “Using Search” describes how to search the contents and attributes of documents on the server. In addition, this chapter describes how to create a customized text search interface that’s tailored to your user community.

Part IV: Managing Virtual Servers and Services

This part provides information for using the Server Manager to programs and configuration styles. The following chapters are included:

- Chapter 13, “Using Virtual Servers” describes how to set up and administer virtual servers using your iPlanet Web Server.
- Chapter 14, “Creating and Configuring Virtual Servers” describes how you can create and configure individual virtual servers.
- Chapter 15, “Extending Your Server With Programs” describes how to install Java applets, CGI programs, JavaScript applications, and other plug-ins onto your server.
- Chapter 16, “Content Management” describes how you can configure and manage your server’s content.
- Chapter 17, “Applying Configuration Styles” describes how to use configuration styles with iPlanet Web Server.

Part V: Appendixes

This section includes various appendixes with reference material that you may wish to review. This section includes the following appendixes:

- Appendix A, “Command Line Utilities” provides instructions for using command line utilities in place of the user interface screens.
- Appendix B, “HyperText Transfer Protocol” provides a short introduction to a few HTTP basic concepts.
- Appendix C, “ACL File Syntax” describes the access-control list (ACL) files and their syntax.
- Appendix D, “Internationalized iPlanet Web Server” describes the internationalized version of the iPlanet Web Server.
- Appendix E, “Server Extensions for Microsoft FrontPage” describes using server extensions on your iPlanet Web Server that provide support for Microsoft FrontPage.

In addition, a glossary is included to define frequently used terms that may be unfamiliar to iPlanet Web Server administrators.

Conventions Used In This Guide

The conventions used in this guide are as follows:

Italic

This typeface is used for book titles, emphasis, and any text that is a placeholder for text you need to replace for your system. For example, in a URL that contains a reference to your server’s port number, the URL might contain *portnumber* in italics. Replace the words in italics with the actual value for your server.

`Monospaced font`

This typeface is used for any text that you should type. It’s also used for functions, examples, URLs, filenames, and directory paths.

Using the iPlanet Web Server Documentation

The following table lists the tasks and concepts that are described in the iPlanet Web Server printed manuals and online README file. If you are trying to accomplish a specific task or learn more about a specific concept, refer to the appropriate manual.

NOTE Printed manuals are also available as online files in PDF and HTML format.

Table 1 iPlanet Web Server Documentation

For information about	See the following
Late-breaking information about the software and the documentation.	http://docs.iplanet.com
Installing iPlanet Web Server.	*iPlanet Web Server Installation & Migration Guide*
Administering one or more iPlanet Web Servers using the Administration Server to manage and configure your servers and to perform the following tasks: • Setting up server security. • Monitoring your servers using HTTP, via log files, SNMP, or via the tools provided with your OS. • Defining your server workload and sizing your system to meet your performance needs. • Installing Java applets, CGI programs, JavaScript applications, and other plug-ins onto your server. • Searching the contents and attributes of server documents; creating a text search interface.	*iPlanet Web Server Administrator's Guide*
The administration server and global information on topics such as encryption, access control, and performance monitoring.	*Managing Servers with Netscape Console*
Planning your directory service. How you can use the directory server to support simple usage that involves only a few hundred users and some key server applications, as well as how you can scale the directory server to support millions of users. You are also introduced to the basic directory service concepts and specific guidelines that you will need to deploy a production-grade directory service.	*iPlanet Directory Server Deployment Manual*
An overview of the programming technologies and APIs you can use to extend and modify iPlanet Web Server, to dynamically generate content in response to client requests, and to modify the content of the server. Links are provided to the individual books that discuss each API. Use this book as the starting place for developer-level information for iPlanet Web Server, Enterprise Edition 6.0. The book also discusses the purpose and use of the configuration files, and provides a comprehensive list of the directives and functions that can be used in these configuration files.	*Programmer's Guide to iPlanet Web Server*
How to enable and implement servlets and JavaServer Pages (JSP) in iPlanet Web Server.	*Programmer's Guide to Servlets in iPlanet Web Server*

Table 1 iPlanet Web Server Documentation *(Continued)*

For information about	See the following
How to use Netscape Server Application Programmer's Interface (NSAPI) to build plugins to extend and modify the iPlanet Web Server. It also provides a reference of the NSAPI functions you can use to define new plugins.	*NSAPI Programmer's Guide for iPlanet Web Server*

Further Reading

The iPlanet documentation site contains documentation for administrators, users, and developers, including:

- iPlanet Web Server *Release Notes*
- Netscape Internet Service Broker programmer's guides and reference guides for Java and C++

To access these documents, use the following URL:

```
http://docs.iplanet.com
```

Contacting Technical Support

For Technical Support assistance, please see the Technical Support Page for the iPlanet Web Server at:

```
http://www.iplanet.com/support/
```

Part 1

Server Basics

Chapter 1, “Introduction to iPlanet Web Server”

Chapter 2, “Administering iPlanet Web Servers”

Chapter 1

Introduction to iPlanet Web Server

This chapter introduces iPlanet Web Server and discusses some of the fundamental server concepts. Read it to obtain an overview of how iPlanet Web Server works.

This chapter includes the following sections:

- iPlanet Web Server
- iPlanet Web Server Architecture
- iPlanet Web Server Configuration
- Administration Server
- Server Manager
- Class Manager
- Virtual Server Manager

iPlanet Web Server

iPlanet Web Server, Enterprise Edition 6.0 is a multi-process, multi-threaded, secure web server built on open standards. It provides high performance, reliability, scalability, and manageability for any size enterprise.

This section includes the following topics:

- iPlanet Web Server Features
- Administering and Managing iPlanet Web Servers

iPlanet Web Server Features

iPlanet Web Server is primarily designed to provide access to your business HTML files. In addition, it offers the following features:

- **Enterprise-wide manageability**—Including delegated administration, cluster management, and LDAP (Lightweight Directory Access Protocol) support. LDAP integration with Directory Server enables you to store users and groups in a centralized directory. In addition, you can monitor your server in real-time by using the *Simple Network Management Protocol* (SNMP). SNMP is a protocol used to exchange data about network activity.Note that in order to add users and groups to iPlanet Web Server, you must have a directory server installed, such as iPlanet Directory Server. See the iPlanet Web Server *Installation and Migration Guide* for more information.**Security**—Users can establish encrypted and authenticated transactions between clients and the server through the Secure Sockets Layer (SSL) 3.0 protocol. In addition, iPlanet Web Server employs the following security-based standards: Public Key Cryptography Standard (PKCS) #11, which defines the interface used for communication between SSL and PKCS #11 modules; Federal Information Processing Standards (FIPS)-140; and special certificates that work with 56, 128, or 168 bits, depending on the capability of the client.
- **Access control**—You can protect confidential files or directories by implementing access control (viewing, editing, and version control) by user name, password, domain name, or IP address. This feature also represents another aspect of the NSAPI Content Management plug-in, which enables an end user (the owner of a document) to set access control on a document, rather than having to ask the administrator to accomplish the task.
- **High performance**—Delivers high performance for dynamic and secure content with features such as HTTP1.1, multi-threading, and support for SSL hardware accelerators.
- **Standards-based**—iPlanet Web Server includes support for a wide range of web software standards, including: JDK 1.2; Servlets 2.2; JavaServer Pages 1.1; HTTP 1.1; and various security-based standards, including PKCS #11, FIPS-140, and 168-bit step-up certificates.
- **Server-side Java Servlet and JavaServer Pages support**—enables development of server plugins, dynamic content, presentation logic, and JDBC database access.
- **Additional features**—Support for multiple processes and process monitors, failover, automatic recovery, and dynamic log rotation.

Administering and Managing iPlanet Web Servers

You can manage your iPlanet Web Server(s) via the following user interfaces:

- iPlanet Web Server Administration Server
- Server Manager
- Class Manager
- Virtual Server Manager

In previous releases, the Server and other Netscape servers were administered by a single server, called the Administration Server. In the 4.x release, the "administration server" became simply an additional instance of the iPlanet Web Server, called iPlanet Web Server Administration Server, or Administration Server. You use the Administration Server to administer all of your iPlanet Web Server instances. For more information, see "Administration Server," on page 38.

NOTE You can also perform administrative tasks manually by editing the configuration files or by using command-line utilities.

For managing individual instances of iPlanet Web Server, you can use the Server Manager. For more information, see "Server Manager," on page 39.

To manage virtual servers, use the Class Manager. For more information, see "Virtual Server Configuration," on page 37.

iPlanet Web Server Architecture

iPlanet Web Server incorporates a modular architecture that integrates seamlessly with all of the products in the iPlanet family of servers. In addition, iPlanet Web Server includes an administration server interface for coordinating administrative functions across all of your web servers. Note that this administrative interface is itself another instance of iPlanet Web Server.

iPlanet Web Server includes the following software modules:

- Content Engines
- Server Extensions

- Runtime Environments
- Application Services

These server modules are described in the following sections.

Content Engines

iPlanet Web Server content engines are designed for manipulating customer data. The following three content engines make up the Web Publishing layer of the iPlanet Web Server architecture: HTTP (Web Server), Content Management, and Search.

The HTTP engine represents the core of the iPlanet Web Server. From a functional perspective, the rest of the iPlanet Web Server architecture resides on top of this engine for performance and integration functionality.

The Content Management engine enables you to manage your server's content. You create and store HTML pages, JavaServer Pages, and other files such as graphics, text, sound, or video on your server. When clients connect to your server, they can view your files provided they have access to them.

The Search engine enables iPlanet Web Server users to search the contents and attributes of documents on the server. As the server administrator, you can create a customized text search interface that works with various types of documents formats, such as HTML, Microsoft Word, Adobe PDF, and WordPerfect. iPlanet Web Server converts many types of non-HTML documents into HTML as it indexes them so that users can use your web browser to view the documents that are found for their search.

Server Extensions

The iPlanet Web Server extensions enable you to extend or replace the function of the server to better suit your business operations. The following server extensions are part of the core iPlanet Web Server architecture:

- Common Gateway Interface (CGI)
- Netscape Server Application Programming Interface (NSAPI)
- Java Servlets and JavaServer Pages

Common Gateway Interface (CGI) is a stand-alone application development interface that enables you to create programs that process your client requests dynamically.

Netscape Server Application Programming Interface (NSAPI) is used to implement the functions the server calls when processing a request (Server Application Functions) which provide the core and extended functionality of the iPlanet Web Server. It allows the server's processing of requests to be divided into small steps which may be arranged in a variety of ways for speed and flexible configuration.

Java Servlets and JavaServer Pages extensions enable all Java servlet and JavaServer page meta-functions, including instantiation, initialization, destruction, access from other components, and configuration management. Java servlets and JavaServer pages, are reusable Java applications that run on a web server rather than in a web browser.

Runtime Environments

In addition to the various server extensions, iPlanet Web Server includes a set of runtime environments which support the server extensions. These runtime environments include the following:

- CGI Processor
- NSAPI Engine
- Java Virtual Machine (JVM)

Application Services

Finally, the iPlanet Web Server architecture includes a set of application services for various application-specific functions. These application services include the following:

- Security & Access Control
- Session Management Service
- File System Service
- Mail Service

iPlanet Web Server Configuration

iPlanet Web Server is configured to enable you to turn on or off various features, determine how to respond to individual client requests, and write programs that run on and interact with the server's operation. The instructions (called directives) which identify these options are stored in configuration files. iPlanet Web Server reads the configuration files on startup and during client requests to map your choices with the desired server activity. For more information about these files, see "iPlanet Web Server Configuration Files," on page 32.

The server includes a number configuration files which are stored in *server_root*/`https-`*server_id*/`config` and *server_root*/`https-admserv/config` when installed on your computer.

This section includes the following topics:

- iPlanet Web Server Component Options
- iPlanet Web Server Configuration Files
- Single-Server Configuration
- Multiple-Server Configuration

iPlanet Web Server Component Options

The following component options are available when you install iPlanet Web Server:

- iPlanet Web Server Core
- Java Runtime Environment
- Java and Servlets

iPlanet Web Server Configuration Files

iPlanet Web Server includes a variety of configuration files that enable you to set various global variables, and to customize how the server responds to specific events and client requests. You can modify the configuration files automatically using the Administrator Server, Server Manager, and Class Manager user interface, or by editing the files directly using a text editor.

The main iPlanet Web Server configuration files are: `magnus.conf`, `obj.conf`, `mime.types`, `server.xml`, and `admpw`. These configuration files are described in this section.

magnus.conf: contains global server configuration information (such as security and default language selection). This file sets the values for variables that configure the server during initialization. iPlanet Web Sever reads this file and executes the variable settings on startup. The server does not read this file again until it is restarted, so you must restart the server every time you make changes to this file.

For more information, see the *NSAPI Programmer's Guide for iPlanet Web Server*.

obj.conf: object configuration file. There is one `obj.conf` file for each virtual server class, or grouping of virtual servers. Whenever this guide refers to "the `obj.conf` file," it refers to all `obj.conf` files or to the `obj.conf` file for the virtual server class being described. All the `obj.conf` files are located in *server_root*/*server_id*/`config` . They are typically named *vsclass*`.obj.conf`, where *vsclass* is the virtual server class name.

The `obj.conf` file contains settings for server customization, and instructions that the server uses to process requests from clients (such as browsers). Each virtual server reads this file every time it processes a client request.

For more information about the actual file syntax and the specific directives used by the `obj.conf` and `magnus.conf` configuration files, see the *NSAPI Programmer's Guide for iPlanet Web Server.*

server.xml: configures the addresses and ports that the server listens on and assigns virtual server classes and virtual servers to these listen sockets. For more information, see the *NSAPI Programmer's Guide for iPlanet Web Server.*

mime.types: the MIME (Multi-purpose Internet Mail Extension) type configuration file. This file maps file extensions to MIME types, to enable the server to determine the type of content being requested. For example, requests for resources with `.html` extensions indicate that the client is requesting an HTML file, while requests for resources with `.gif` extensions indicate that the client is requesting an image file in GIF format.

For more information, see "Specifying a Default MIME Type," on page 352 in Chapter 16, "Content Management." Note that you must restart the server every time you make changes to this file.

admpw: the user name and password file for the Administrator Server superuser. For more information, see "Changing the Superuser Settings," on page 51 in Chapter 3, "Setting Administration Preferences."

Dynamic Reconfiguration

Dynamic reconfiguration allows you to make configuration changes to a live web server without having to stop and restart the web server for the changes to take effect. You can dynamically change all configuration settings and attributes in `server.xml` and its associated files without restarting the server.

To access the dynamic reconfiguration screen and install a new configuration dynamically, click the Apply link found in the upper right corner of the Server Manager, Class Manager, and Virtual Server Manager pages, then click the Load Configuration Files button on the Apply Changes page. If there are errors in installing the new configuration, the previous configuration is restored.

Single-Server Configuration

If you have installed iPlanet Web Server on a single server machine, the installation process places all the files under the server root directory that you specified during installation.

All Platforms

For all platforms, the following directories are created under the server root directory:

- **alias** contains the key and certificate files for all iPlanet servers (for example, `https-admserv-`*`server_id-cert7.db`* and `secmod.db`).
- **bin** contains the binary files for the server, such as the actual server, the Administration Server forms, and so on. In addition, this directory includes the `https/install` folder that contains files needed for migrating server settings and default configuration files needed for backward compatibility.
- **docs** is the server’s default primary document directory, where your server’s content files are usually kept. If you are migrating settings from an existing server, this directory doesn’t appear until you finish the migration process.
- **extras** contains the log analyzer and log analysis tools.
  - The `flexanlg` directory contains a command-line log analyzer. This log analyzer analyzes files in flexlog format.
  - The `log_anly` directory contains the log analysis tool that runs through the Server Manager. This log analyzer analyzes files in common log format only.

- **httpacl** contains the files that store access control configuration information in the `generated.`*server-id*`.acl` and `genwork.`*server-id*`.acl` files. The file `generated.`*server-id*`.acl` contains changes you make using the Server Manager access control forms after saving your changes; `genwork.`*server-id*`.acl` contains your changes *before* you save your changes.
- **https-admserv** contains the directories for the Administration Server. This directory has the following subdirectories and files:
  - For Unix/Linux platforms, this directory contains shell scripts to start, stop, and restart the server and a script to rotate log files.
  - `ClassCache` contains classes and Java files, generated as result of the compilation of JavaServer pages.
  - `conf_bk` contains backup copies of the administration server's configuration files.
  - `config` contains the server's configuration files: `admpw, admin.conf, cluster.xml, contexts.properties, cron.conf, dsgw.conf, dsgwfilter.conf, dsgw-language.conf, dsgw-orgperson.conf, dsgwserarchprefs.conf, jvm12.conf, magnus.conf, magnus.conf.clfilter, mime.types, ns-cron.conf, obj.conf, obj.conf.clfilter, server.dtd, servers.lst, server.xml, server.xml.clfilter`, `servlets.properties, ssl.xml, user-apps.xml, userclass.obj.conf`, and `web-apps.xml.` Working copies are kept here. For more information on `magnus.conf` and `obj.conf`, see the *NSAPI Programmer's Guide for iPlanet Web Server.*
  - `logs` contains any error or access log files.
  - `SessionData` contains session database data from `MMapSessionManager`.
  - `startsvr.bat` is the script that starts the Server Manager on Windows NT machines. The Server Manager lets you configure all servers installed in the server root directory.
  - `stopsvr.bat` is the script that stops the Server Manager on Windows NT machines.
- **https-server_id.domain** are the directories for each server you have installed on the machine. Each server directory has the following subdirectories and files:
  - `ClassCache` contains classes and Java files, generated as result of the compilation of JavaServer pages.
  - `conf_bk` contains backup copies of the server's configuration files.
  - `config` contains the server instance configuration files.

  - ❍ logs contains the server instance log files.
  - ❍ reconfig is the script used to reconfigure the server dynamically. If you make non-global changes to the server, you can use this script to reconfigure the server without stopping and starting it. Note that changes to ACL files and magnus.conf require you to stop and restart the server.
  - ❍ restart is the script that restarts the server.
  - ❍ rotate rotates server log files without affecting users who may be connected to the server.
  - ❍ search contains the following directories: admin and collections
  - ❍ SessionData contains session database data from MMapSessionManager.
  - ❍ startsvr.bat is the script that starts the Server Manager. The Server Manager lets you configure all servers installed in the server root directory.
  - ❍ stopsvr.bat is the script that stops the Server Manager.
- **manual** contains the online manuals for the product.
- **plugins** contains directories for Java, search, and other plugins. This directory has the following subdirectories:
  - ❍ htaccess contains server plugin for .htaccess access control and htconvert, an .nsconfig to .htaccess converter.
  - ❍ digest contains the Digest Authentication Plugin for iPlanet Directory Server 5.0, as well as information about the plugin.
  - ❍ samples contains samples and example components, plugins and technologies supported by the iPlanet Web Server servlet engine. This includes binaries, all code, and a build enviroment.
  - ❍ servlets contains information about and examples of web-apps applications.
  - ❍ include contains various include files.
  - ❍ lib contains shared libraries.
  - ❍ nsacl contains information for your server's access control lists.
  - ❍ loadbal contains the required files for the Resonate load-balancer integration plugin.
  - ❍ nsapi contains header files and example code for creating your own functions using NSAPI. For more information, see the iPlanet documentation web site at: http://docs.iplanet.com/docs/manuals/enterprise.html.

  - `search` contains information for your server's search plugins.
  - `snmp` contains information for your server's SNMP plugins.

- **setup** contains the various iPlanet Web Server setup files, including `setup.log` and `uninstall.inf`.
- **userdb** contains user databases and related information.
- **LICENSE.txt** is the license file.
- **README.txt** is the readme file that contains a link to the iPlanet Web Server *Notes*.

Unix and Linux Platforms

In addition to the files and directories described in "All Platforms," the following files are created at the `server-root` directory for Unix and Linux platforms:

- **startconsole** launches a browser to the Administration Server page.

The following files are created under the *server-root*/`https-admserv` directory for Unix and Linux platforms:

  - `ClassCache` contains classes and Java files, generated as result of the compilation of JavaServer pages.
  - `conf_bk` contains backup copies of the server's configuration files.
  - `config` contains the Administration Server configuration files.
  - `logs` contains the Administration Server log files.
  - `SessionData` contains session database data from `MMapSessionManager`.
  - `restart` is the script that restarts the Server Manager.
  - `start` is the script that starts the Server Manager. The Server Manager lets you configure all servers installed in the server root directory.
  - `stop` is the script that stops the Server Manager.

Virtual Server Configuration

Virtual servers allow you, with a single installed server, to offer companies or individuals domain names, IP addresses, and some server administration capabilities. You can configure virtual servers using the Virtual tab of the Server Manager, as well as the Class Manager interface and the `server.xml` file. The settings for virtual servers are stored in the `server.xml` file, found in the *server_root/server_id/*`config` directory.

For more information, see Chapter 13, “Using Virtual Servers.”

Multiple-Server Configuration

You can have multiple web servers running on the same server machine. Multiple web servers can be configured from a single-server administration interface called the Administration Server.

Administration Server

The Administration Server is a web-based server that contains the Java forms you use to configure all of your iPlanet Web Servers.

After installing iPlanet Web Server, you use your browser to navigate to the Administration Server page and use its forms to configure your iPlanet Web Servers. When you submit the forms, the Administration Server modifies the configuration for the server you were administering.

The URL you use to navigate to the Administration Server page depends on the computer host name and the port number you choose for the Administration Server when you install iPlanet Web Server. For example, if you installed the Administration Server on port 1234, the URL would look like this:

```
http://myserver.mozilla.com:1234
```

Before you can get to any forms, the Administration Server prompts you to authenticate yourself. This means you need to type a user name and password. You set up the “superuser” user name and password when you install iPlanet Web Server on your computer. After installation, you can use distributed administration to give multiple people access to different forms in the Administration Server. For more information about distributed administration, see “Allowing Multiple Administrators,” on page 53 in Chapter 3, “Setting Administration Preferences.”

The first page you see when you access the Administration Server, is called Servers. You use the buttons on this page to manage, add, remove, and migrate your iPlanet Web Servers. The Administration Server provides the following tabs for your administration-level tasks:

- Servers
- Preferences
- Global Settings
- Users and Groups

- Security
- Cluster Mgmt (Cluster Management)

NOTE You must enable cookies in your browser to run the CGI programs necessary for configuring your server.

For more information on using the Administration Server, including information regarding these administration-level tasks, see Chapter 2, “Administering iPlanet Web Servers.”

Server Manager

The Server Manager is a web-based interface that contains the Java forms you use to configure individual instances of iPlanet Web Server.

You can access the Server Manager for iPlanet Web Server by performing the following steps:

1. Install and start your iPlanet Web Server.

   The Administration Server displays the Servers page.

2. In the Manage Servers area, select the desired server and click Manage.

   iPlanet Web Server displays the Server Manager Preferences page.

NOTE Note that you must enable cookies in your browser to run the CGI programs necessary for configuring your server.

You use the links on the Preferences page to manage options such as thread pool settings, and to turn the web server on and off.

In addition, the Server Manager provides the following tabs for additional iPlanet Web Server managerial tasks:

- Security
- Logs
- Monitor
- Virtual Server Class
- Java

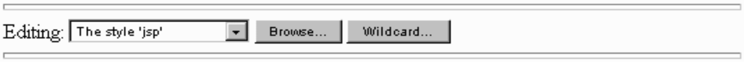
- Legacy Servlets
- Search

For more information, see the Server Manager in the online help.

Using the Resource Picker

Most of the Server Manager and Class Manager pages configure the entire iPlanet Web Server or an entire class. However, some pages can configure either the entire server (or class) or files and directories that the server (or class) maintains. These pages include the Resource Picker, shown in Figure 1-1, at the top.

Figure 1-1 Resource Picker



The Resource Picker appears on a number of pages, including the Server Manager's Log Preferences page and most screens accessible from the Class Manager's Content Management tab.

To use the Resource Picker, choose a resource from the drop-down list for configuration. Click Browse to browse your primary document directory; clicking Options allows you to choose other directories. Click Wildcard to configure files with a specific extension.

Wildcards Used in the Resource Picker

In many parts of the server configuration, you specify wildcard patterns to represent one or more items to configure. Please note that the wildcards for access control and text search may be different from those discussed in this section.

Wildcard patterns use special characters. If you want to use one of these characters without the special meaning, precede it with a backslash (\) character.

Class Manager

The Class Manager is a web-based interface that contains the Java forms you use to configure your virtual iPlanet Web Servers. The user interface for virtual servers has two parts, the Server Manager and the Class Manager. The Class Manager contains settings that affect a single class or single virtual server. You can set services for the class in the Class Manager, as well as add virtual servers (members of the class) and configure settings for an individual virtual server.

You can access the Class Manager for iPlanet Web Server by performing the following steps:

1. From the Server Manager, click the Virtual Server Class tab.

   The Server Manager displays the Select a Class of Virtual Server page.

2. From the drop-down list, select a virtual server class and click Manage.

   iPlanet Web Server displays the Class Manager’s Select a Virtual Server page.

You can also access the Class Manager by simply clicking the Class Manager link in the upper right-hand corner of the screen.

The Class Manager provides the following tabs to manage your iPlanet Web Server virtual servers:

- Virtual Servers
- Programs
- Content Management
- Styles

For more information, see the Class Manager in the online help.

Virtual Server Manager

To access the Virtual Server Manager, go to the Virtual Servers tab in the Class Manager, then select a virtual server from the list on the Manager Virtual Servers page and click Manage, or click on the link to a virtual server under the tree view.

The pages provided in the Virtual Server Manager allow you to check the status and settings, set the Java web applications state to on, and generate reports for the selected virtual server.

The Virtual server Manager provides the following tabs to manage your iPlanet Web Server virtual servers:

- Preferences
- Logs
- Web Applications

Chapter 2

Administering iPlanet Web Servers

This chapter describes how to administer iPlanet Web Server, Enterprise Edition 6.0 with the iPlanet Web Server Administration Server. Using the Administration Server, you can manage servers, add and remove servers, and migrate servers from a previous release.

This chapter includes the following sections:

- Accessing the Administration Server
- Running Multiple Servers
- Installing Multiple Instances of the Server
- Removing a Server
- Migrating a Server From a Previous Version

Accessing the Administration Server

This section describes how to access the Administration Server for Unix/Linux and Windows NT platforms.

Unix/Linux Platforms

To access the Administration Server in Unix or Linux platforms, go to the `server_root/https-admserv/` directory (for example, `/usr/iplanet/servers/https-admserv/`) and type `./start`. This command starts the Administration Server using the port number you specified during installation.

Windows NT Platforms

The iPlanet Web Server installation program creates a program group with several icons for Windows NT platforms. The program group includes the following icons:

- Release Notes
- Start Administration Server
- Uninstall iPlanet Web Server 6.0
- Administer Web Server

Note that the Administration Server runs as a services applet; thus, you can also use the Control Panel to start this service directly.

To access the Administration Server in Windows NT 4.0, perform the following steps:

1. Double-click the “Start Administration Server” icon, or type the following URL for starting the administration server in your browser:

   `http://hostname.domain-name:administration_port`

   iPlanet Web Server then displays a window prompting you for a user name and password.

2. Type the administration user name and password you specified during installation.

   iPlanet Web Server displays the Administration Server page.

For more information, see Administration Server. in the online help.

NOTE You must enable cookies in your browser to run the CGI programs necessary for configuring your server.

You can also access the Administration Server from a remote location as long as you have access to client software such as Netscape Navigator. Since the Administrator Server is accessed through a browser, you can access it from any machine that can reach the server over the network.

Running Multiple Servers

There are two ways you can have multiple web servers running on your system:

- Use virtual servers
- Install multiple instances of the server

Virtual Servers

Virtual servers allow you, with a single installed server, to offer companies or individuals domain names, IP addresses, and some server administration capabilities. For the users, it is almost as if they have their own web server, though you provide the hardware and basic web server maintenance.

The settings for virtual servers are stored in the `server.xml` file, found in the *server_root/server_id/*`config` directory. You do not need to edit this file to use virtual servers, but if you would like to learn more about this file, see the *NSAPI Programmer's Guide*.

For more information about virtual servers, see Chapter 13, "Using Virtual Servers."

Installing Multiple Instances of the Server

In past releases of iPlanet Web Server, virtual servers did not have unique configuration information. The only way to have servers with separate configuration information was to create a new server instance. However, with iPlanet Web Server, Enterprise Edition 6.0, virtual servers have separate configuration information, so multiple server instances are no longer required. They are still supported, but virtual servers are the preferred way to have multiple servers.

If you choose to install multiple instances of the web server, you can use the Administration Server to:

- Install multiple copies of the server on NT as separate instances, each with a different IP address.
- Configure a set of servers that all use the same IP address, but different port numbers.

If your system is configured to listen to multiple IP addresses enter one of the IP addresses that your system is hosting for each server you install.

If you installed your server before configuring your system to host multiple IP addresses, configure your system to respond to different IP addresses. Then you can either install hardware virtual servers or change the server's bind address using the Server Manager and install separate instances of the server for each IP address.

To add another server instance, perform the following steps:

1. Access the Administration Server and choose the Servers tab.
2. Click the Add Server link.

3. Enter the desired information for the specified fields.

   Note that the server identifier cannot start with a digit and only Latin-1 characters should be used in instance names.

For more information, see The Add Server Page in the online help.

Removing a Server

You can remove a server from your system using the Administration Server. Be sure that you don't need the server anymore before you remove it, since this process cannot be undone.

NOTE Some NT servers have an uninstall program that you can use to remove a server and its associated administration server. For details, check with your product documentation.

To remove a server from your machine, perform the following steps:

1. Access the Administration Server and choose the Servers tab.
2. Click Remove Server.

The Administration Server subsequently deletes the server's configuration files, Server Manager forms, and the following directory (and any subdirectories):

server_root/`https-`*server-id*

For more information, see The Remove Server Page in the online help.

Migrating a Server From a Previous Version

You can migrate an Server from 4.x to 6.0. Your 4.x server is preserved, and a new 6.0 server using the same settings is created.

You should stop running the 4.x server before migrating settings. Make sure you have a compatible version of a web browser installed on your computer before migrating settings.

For a complete description of how to migrate a server from a previous version to , see the *Installation and Migration Guide.*

For more information, see The Migrate Server Page in the online help.

Part 2

Using the Administration Server

Chapter 3

Setting Administration Preferences

You can configure your Administration Server using the pages on the Preferences and Global Settings tabs. Note that you must enable cookies in your browser to run the CGI programs necessary for configuring your server.

This chapter includes the following sections:

- Shutting Down the Administration Server
- Editing Listen Socket Settings
- Changing the User Account (Unix/Linux)
- Changing the Superuser Settings
- Allowing Multiple Administrators
- Specifying Log File Options
- Configuring Directory Services
- Restricting Server Access
- Configuring JRE/JDK Paths

Shutting Down the Administration Server

Once the server is installed, it runs constantly, listening for and accepting HTTP requests. You might want to stop and restart your server if, for instance, you have just installed a Java Development Kit (JDK) or Directory Server, or if you have changed listen socket settings.

You can stop the server using one of the following methods:

- Access the Administration Server, choose the Preferences tab, select the Shut Down link, and click "Shut down the administration server button!".

  For more information, see The Shut Down Page in the online help.

- Use the Services window in the Control Panel (Windows NT).
- Use `stop`, which shuts down the server completely, interrupting service until it is restarted.

After you shut down the server, it may take a few seconds for the server to complete its shut-down process and for the status to change to "Off."

Editing Listen Socket Settings

Before the server can process a request, it must accept the request via a listen socket, then direct the request to the correct connection group and virtual server. When you install iPlanet Web Server, one listen socket, ls1, is created automatically. This listen socket uses the IP address 0.0.0.0 and the port number you specified as your HTTP server port number during installation (the default is 8888). You cannot delete the default listen socket.

You can edit your server's listen socket settings using the Administration Server's Listen Sockets Table. To access the table, perform the following steps:

1. Access the Administration Server and click the Preferences tab.
2. Click the Edit Listen Sockets link.
3. Make the desired changes and click OK.

For more information, see Chapter 13, "Using Virtual Servers" and the online help for The Edit Listen Sockets Page.

Changing the User Account (Unix/Linux)

The Server Settings page allows you to change the user account for your web server on Unix and Linux machines. All the server's processes run as this user.

You do not need to specify a server user if you chose a port number greater than 1024 and are not running as the `root` user (in this case, you do not need to be logged on as `root` to start the server). If you do not specify a user account here, the server runs with the user account you start it with. Make sure that when you start the server, you use the correct user account.

NOTE If you do not know how to create a new user on your system, contact your system administrator or consult your system documentation.

Even if you start the server as root, you should not run the server as root all the time. You want the server to have restricted access to your system resources and run as a non-privileged user. The user name you enter as the server user should already exist as a normal Unix/Linux user account. After the server starts, it runs as this user.

If you want to avoid creating a new user account, you can choose the user `nobody` or an account used by another HTTP server running on the same host. On some systems, however, the user `nobody` can own files but not run programs.

To access the Server Settings page, perform the following steps:

1. Access the Administration Server and choose the Preferences tab.
2. Click the Server Settings link.
3. Make the desired changes and click OK.

Changing the Superuser Settings

You can configure superuser access for your Administration Server. These settings affect only the superuser account. That is, if your Administration Server uses distributed administration, you need to set up additional access controls for the administrators you allow.

CAUTION If you use iPlanet Directory Server to manage users and groups, you need to update the superuser entry in the directory *before* you change the superuser user name or password. If you don’t update the directory first, you won’t be able to access the Users & Groups forms in the Administration Server. To fix this, you’ll need to either access the Administration Server with an administrator account that does have access to the directory, or you’ll need to update the directory using the iPlanet Directory Server’s Console or configuration files.

To change the superuser settings for the Administration Server, perform the following steps:

1. Access the Administration Server and choose the Preferences tab.
2. Click the Superuser Access Control link.
3. Make the desired changes and click OK..

NOTE You can change the Administration Server user from root to another user on the operating system to enable multiple users (belonging to the group) to edit/manage the configuration files. However, note that while on Unix/Linux platforms, the installer can give "rw" (read/write) permissions to a group for the configuration files, on Windows NT platforms, the user must belong to the "Administrators" group.

The superuser's user name and password are kept in a file called *server_root*/https-admserv/config/admpw. If you forget the user name, you can view this file to obtain the actual name; however, note that the password is encrypted and unreadable. The file has the format username:password. If you forget the password, you can edit the admpw file and simply delete the encrypted password. You can then go to the Server Manager forms and specify a new password.

CAUTION Because you can edit the admpw file, it is very important that you keep the server computer in a secure place and restrict access to its file system:

- On Unix/Linux systems, consider changing the file ownership so that it's writable only by root or whatever system user runs the Administration Server daemon.
- On NT systems, restrict the file ownership to the user account Administration Server uses.

Allowing Multiple Administrators

Multiple administrators can change specific parts of the server through distributed administration. With distributed administration you have three levels of users:

- **superuser** is the user listed in the file `*server_root*/https-admserv/config/admpw`. This is the user name (and password) you specified during installation. This user has full access to all forms in the Administration Server, except the Users & Groups forms, which depend on the superuser having a valid account in an LDAP server such as iPlanet Directory Server.
- **administrators** go directly to the Server Manager forms for a specific server, including the Administration Server. The forms they see depend on the access control rules set up for them (usually done by the superuser). Administrators can perform limited administrative tasks and can make changes that affect other users, such as adding users or changing access control.
- **end users** can view read-only data stored in the database. Additionally, end users may be granted access permissions to change only specific data.

For an in-depth discussion of access control for iPlanet Web Server, see “What Is Access Control?,” on page 154 in Chapter 8, “Controlling Access to Your Server.”

NOTE Before you can enable distributed administration, you must install a Directory Server. For more information, see the iPlanet Web Server *Installation and Migration Guide* and the iPlanet Directory Server *Administrator’s Guide*.

To enable distributed administration, perform the following steps:

1. Verify that you have installed a Directory Server.
2. Access the Administration Server.
3. One you’ve installed a Directory Server, you may also need to create an administration group, if you have not previously done so.

   To create a group, perform the following steps:

   a. Choose the Users & Groups tab.

   b. Click the New Group link.

c. Create an “administrators” group in the LDAP directory and add the names of the users you want to have permission to configure the Administration Server, or any of the servers installed in its server root. All users in the “administrators” group have full access to the Administration Server, but you can use access control to limit the servers and forms they will be allowed to configure.

CAUTION Once you create an access-control list, the distributed administration group is added to that list. If you change the name of the “administrators” group, you must manually edit the access-control list to change the group it references.

4. Choose the Preferences tab.
5. Click the Distributed Admin link.
6. Make the desired changes and click OK.

For more information, see The Distributed Administration Page in the online help.

Specifying Log File Options

The Administration Server log files record data about the server, including the types of errors encountered and information about server access. Viewing these logs allows you to monitor server activity and troubleshoot problems by providing data like the type of error encountered and the time certain files were accessed.

You can specify the type and format of the data recorded in the Administration Server logs using the Log Preferences page. For instance, you can choose to log data about every client who accesses the Administration Server or you can omit certain clients from the log. In addition, you can choose the Common Logfile Format, which provides a fixed amount of information about the server, or you can create a custom log file format that better suits your requirements.

Access the Administration Server Log Preferences page by choosing the Preferences tab, then clicking the Logging Options link.

For more information, see The Logging Options Page in the online help, and Chapter 9, “Using Log Files.”

Viewing Log Files

The Administration Server log files are located in `admin/logs` in your server root directory. For example, on Windows NT, the path to your log files might look like `c:\iPlanet\server6\https-admserv\logs`. You can view both the error log and the access log through the iPlanet Web Server console or using a text editor.

The Access Log File

The access log records information about requests to and responses from the server.

To view the access log file, perform the following steps:

1. Access the Administration Server and choose the Preferences tab.
2. Click the View Access Log link and click OK.

For more information, see The View Error Log Page in the online help, and Chapter 9, “Using Log Files.”

The Error Log File

The error log lists all the errors the server has encountered since the log file was created. It also contains informational messages about the server, such as when the server was started and who tried unsuccessfully to log in to the server.

To view the error log file, perform the following steps:

1. Access the Administration Server and choose the Preferences tab.
2. Click the View Error Log link and click OK.

For more information, see The View Access Log Page in the online help, and Chapter 9, “Using Log Files.”

Archiving Log Files

You can set up your log files to be automatically archived. At a certain time, or after a specified interval, iPlanet Web Server rotates your access logs. iPlanet Web Server saves the old log files and stamps the saved file with a name that includes the date and time they were saved.

For example, you can set up your files to rotate every hour, and iPlanet Web Server saves and names the file “`access.199907152400`,” where “name|year|month|day|24-hour time” is concatenated together into a single character string. The exact format of the access log archive file varies depending upon which type of log rotation you set up.

Access log rotation is initialized at server startup. If rotation is turned on, iPlanet Web Server creates a time-stamped access log file and rotation starts at server startup.

Once the rotation starts, iPlanet Web Server creates a new time stamped access log file when there is a request that needs to be logged to the access log file and it occurs after the previously-scheduled “next rotate time.”

Using Cron-based Log Rotation (Unix/Linux)

You can configure several features of your iPlanet Web Server to operate automatically and set to begin at specific times. The cron daemon checks the computer clock and then spawns processes at certain times. (These settings are stored in the `ns-cron.conf` file.)

This cron daemon controls scheduled tasks for your iPlanet Web Server and can be activated and deactivated from the Administration Server. The tasks performed by the cron process depends on the various servers. (Note that on NT platforms, the scheduling occurs within the individual servers.)

Some of the tasks that can be controlled by cron daemons include scheduling collection maintenance and archiving log files. You need to restart cron control whenever you change the settings for scheduled tasks.

To restart, start, or stop cron control, perform the following steps:

1. Access the Administration Server and choose the Global Settings tab.
2. Click the Cron Control link.
3. Click Restart, Start, or Stop to change the cron controls.

Note that any time you add a task to cron, you need to restart the daemon.

Configuring Directory Services

You can store and manage information such as the names and passwords of your users in a single Directory Server using an open-systems server protocol called the Lightweight Directory Access Protocol (LDAP). You can also configure the server to allow your users to retrieve directory information from multiple, easily accessible network locations.

To configure the directory services preferences, perform the following steps:

1. Access the Administration Server and choose the Global Settings tab.
2. Click the Configure Directory Service link.
3. Make the desired changes and click OK.

For more information, see The Configure Directory Service Page in the online help.

Restricting Server Access

You can control access to the entire server or to parts of the server (that is, directories, files, file types). When the server evaluates an incoming request, it determines access based on a hierarchy of rules called access-control entries (ACEs), and then it uses the matching entries to determine if the request is allowed or denied. Each ACE specifies whether or not the server should continue to the next ACE in the hierarchy. The collection of ACEs is called an access-control list (ACL). When a request comes in to the server, the server looks in *vsclass*`.obj.conf` (where *vsclass* is the virtual server class name) for a reference to an ACL, which is then used to determine access. By default, the server has one ACL file that contains multiple ACLs.

You can set access control globally for all servers through the Administration Server or for a resource within a specific server instance through the Server Manager. For more information about setting access control for a resource, see “Setting Access Control,” on page 164 in Chapter 8, “Controlling Access to Your Server.”

NOTE You must turn on distributed administration before you can restrict server access.

To restrict access to your iPlanet Web Servers, perform the following steps:

1. Access the Administration Server and choose the Global Settings tab.
2. Click the Restrict Access link.
3. Select the desired server and click Edit ACL.

   The Administration Server displays the access control rules for the server you specified.

Make the desired access control changes and click OK. For more information, see The Restrict Access Page in the online help.

Configuring JRE/JDK Paths

When you install iPlanet Web Server, you can choose to install the Java Runtime Environment (JRE), which is bundled with iPlanet Web Server, or you can specify a path to the Java Development Kit (JDK), which you must install separately. See the iPlanet Web Server *Installation and Migration Guide* for more information.

Regardless of whether you choose to install the JRE or specify a path to the JDK during installation, you can tell the iPlanet Web Server to switch to using either the JRE or JDK at any time by performing the following steps:

1. Access the iPlanet Web Server Administration Server.
2. Select the Global Settings tab.
3. Click the Configure JRE/JDK Paths link.

   The Configure JRE/JDK Paths page appears.
4. Click the radio button corresponding to the feature to enable.

   For instance, click JDK to supply the path to the Java Development Kit installed on your machine.
5. Enter the appropriate information and click OK.

   You must restart your server for changes to become effective.

   See The Configure JRE/JDK Paths Page in the online help for more information.

Chapter 4

Managing Users and Groups

This chapter describes how to add, delete, and edit the users and groups who can access your iPlanet Web Server.

This chapter includes the following sections:

- Using LDAP to Manage Users and Groups
- Creating Users
- Managing Users
- Creating Groups
- Managing Groups
- Creating Organizational Units
- Managing Organizational Units
- Managing a Preferred Language List

Using LDAP to Manage Users and Groups

The Administration Server provides access to your application data about user accounts, group lists, access privileges, organization units, and other user- and group-specific information.

User and group information is stored in a directory server such as iPlanet Directory Server 5.0, which supports Lightweight Directory Access Protocol (LDAP). LDAP is an open directory access protocol that runs over TCP/IP and is scalable to a global size and millions of entries.

Since iPlanet Web Server does not support local LDAP, you must have a directory server installed before you can add users and groups.

Understanding Distinguished Names (DNs)

Use the Users and Groups tab of the Administration Server to create or modify users, groups, and organizational units. A user is an individual in your LDAP database, such as an employee of your company. A group is two or more users who share a common attribute. An organizational unit is a subdivision within your company that uses the `organizationalUnit` object class. Users, groups, and organizational units are described further later in this chapter.

Each user and group in your enterprise is represented by a Distinguished Name (DN) attribute. A DN attribute is a text string that contains identifying information for an associated user, group, or object. You use DNs whenever you make changes to a user or group directory entry. For example, you need to specify DN information each time you create or modify directory entries, set up access controls, and set up user accounts for applications such as mail or publishing. The users and groups interface of Netscape Console helps you create or modify DNs.

The following example represents a typical DN for an employee of Netscape Communications Corporation:

```
uid=doe,e=doe@netscape.com,cn=John Doe,o=Netscape Communications
Corp.,c=US
```

The abbreviations before each equal sign in this example have the following meanings:

- `uid`: user ID
- `e`: email address
- `cn`: the user's common name
- `o`: organization
- `c`: country

DNs may include a variety of name-value pairs. They are used to identify both certificate subjects and entries in directories that support LDAP.

Using LDIF

If you do not currently have a directory, or if you want to add a new subtree to an existing directory, you can use the Directory Server's Administration Server LDIF import function. This function accepts a file containing LDIF and attempts to build a directory or a new subtree from the LDIF entries. You can also export your current directory to LDIF using the Directory Server's LDIF export function. This function creates an LDIF-formatted file that represents your directory. Add or edit entries using the `ldapmodify` command along with the appropriate LDIF update statements.

To add entries to the database using LDIF, first define the entries in an LDIF file, then import the LDIF file from Directory Server. For more information, see "Formatting LDIF Entries," on page 369 of Appendix A, "Command Line Utilities."

Creating Users

Use the Users and Groups tab of the Administration Server to create or modify user entries. A user entry contains information about an individual person or object in the database.

This section includes the following topics:

- Guidelines for Creating User Entries
- How to Create a New User Entry
- Directory Server User Entries

Guidelines for Creating User Entries

Consider the following guidelines when using the administrator forms to create new user entries:

- If you enter a given name (or first name) and a surname, then the form automatically fills in the user's full name and user ID for you. The user ID is generated as the first initial of the user's first name followed by the user's last name. For example, if the user's name is Billie Holiday, then the user ID is automatically set to *bholiday*. You can replace this user ID with an ID of your own choosing if you wish.
- The user ID must be unique. The Administration Server ensures that the user ID is unique by searching the entire directory from the search base (base DN) down to see if the user ID is in use. Be aware, however, that if you use the Directory Server `ldapmodify` command line utility (if available) to create a user, that it does not ensure unique user IDs. If duplicate user IDs exist in your directory, the affected users will not be able to authenticate to the directory.
- Note that the base DN specifies the distinguished name where directory lookups will occur by default, and where all iPlanet Web Administration Server's entries are placed in your directory tree. A "DN" is the string representation for the name of an entry in a directory server.
- Note that at a minimum, you must specify the following user information when creating a new user entry:
    - surname or last name
    - full name
    - user ID
- If any organizational units have been defined for your directory, you can specify where you want the new user to be placed using the Add New User To list. The default location is your directory's base DN (or root point).

NOTE The user edit text fields for international information differs between the Administration Server and Netscape Console. In Netscape Console, in addition to the untagged `cn` fields, there is a preferred language `cn` field which doesn't exist in the Administration Server.

How to Create a New User Entry

To create a user entry, read the guidelines outlined in “Guidelines for Creating User Entries,” on page 62, then perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the New User link and add the associated information to the displayed page.

For more information, see The New User Page in the online help.

Directory Server User Entries

The following user entry notes may be of interest to the directory administrator:

- User entries use the `inetOrgPerson`, `organizationalPerson`, and `person` object classes.
- By default, the distinguished name for users is of the form:

  ```
  cn=full name, ou=organization, ...,o=base organization, c=country
  ```

  For example, if a user entry for Billie Holiday is created within the organizational unit Marketing, and the directory’s base DN is o=Ace Industry, c=US, then the person’s DN is:

  ```
  cn=Billie Holiday, ou=Marketing, o=Ace Industry, c=US
  ```

  However, note that you can change this format to a uid-based distinguished name.
- The values on the user form fields are stored as the following LDAP attributes (note that any stored information other than ‘user’ and ‘group’ requires a full Directory Server license):

Table 4-1 LDAP Attributes

User Field	Corresponding LDAP Attribute
Given Name	`givenName`
Surname	`sn`
Full Name	`cn`
User ID	`uid`
Password	`userPassword`
Email Address	`mail`

The following fields are also available when editing the user entry:

Table 4-2 User Entry LDAP Attributes

User Field	Corresponding LDAP Attribute
Title	`title`
Telephone	`telephoneNumber`

- Sometimes a user's name can be more accurately represented in characters of a language other than the default language. You can select a preferred language for users so that their names will display in the characters of the that language, even when the default language is English. For more information regarding setting a user's preferred language, see The Manage Users Page in the online help.

Managing Users

You edit user attributes from the Administration Server Manage Users form. From this form you can find, change, rename, and delete user entries; manage user licenses; and potentially change product-specific information.

Some, but not all, Netscape/iPlanet servers add additional forms to this area that allow you to manage product-specific information. For example, if a messaging server is installed under your Administration Server, then an additional form is added that allows you to edit messaging server-specific information. See the server documentation for details on these additional management capabilities.

This section includes the following topics:

- Finding User Information
- Editing User Information
- Managing a User's Password
- Managing User Licenses
- Renaming Users
- Removing Users

Finding User Information

Before you can edit a user entry, you must display the associated information. To find the specific user information, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the Manage Users link.
3. In the Find User field, enter some descriptive value for the entry that you want to edit. You can enter any of the following in the search field:

    - A name. Enter a full name or a partial name. All entries that equally match the search string will be returned. If no such entries are found, all entries that contain the search string will be found. If no such entries are found, any entries that sounds like the search string are found.
    - A user ID.
    - A telephone number. If you enter only a partial number, any entries that have telephone numbers ending in the search number will be returned.
    - An email address. Any search string containing an at (@) symbol is assumed to be an email address. If an exact match cannot be found, then a search is performed to find all email addresses that begin with the search string.
    - An asterisk (*) to see all of the entries currently in your directory. You can achieve the same effect by simply leaving the field blank.
    - Any LDAP search filter. Any string that contains an equal sign (=) is considered a search filter.

    As an alternative, use the pull down menus in the Find all users whose field to narrow the results of your search.

4. In the Look within field, select the organizational unit under which you want to search for entries.

    The default is the directory's root point (or top most entry).

5. In the Format field, choose either On-Screen or Printer.
6. Click Find.

    All the users in the selected organizational unit are displayed.

7. In the resulting table, click the name of the entry that you want to edit.

   The user edit form is displayed.

8. Change the displayed fields as desired and click Save Changes.

   The changes are made immediately.

Building Custom Search Queries

The "Find all users whose" field allows you to build a custom search filter. Use this field to narrow down the search results returned by a "Find user" search.

The Find all users whose field provides the following search criteria:

- The left-most pull-down list allows you to specify the attribute on which the search will be based.

The available search attribute options are described in the following table:

Table 4-3 Search Attribute Options

Option Name	Description
full name	Search each entry's full name for a match.
last name	Search each entry's last name, or surname for a match.
user id	Search each entry's user id for a match.
phone number	Search each entry's phone number for a match.
email address	Search each entry's email address for a match.
unit name	Search each entry's name for a match.
description	Search each organizational unit entry's description for a match.

- In the center pull-down list, select the type of search you want to perform.

The available search type options are described in the following table:

Table 4-4 Search Type Options

Option Name	Description
`contains`	Causes a substring search to be performed. Entries with attribute values containing the specified search string are returned. For example, if you know an user's name probably contains the word "Dylan," use this option with the search string "Dylan" to find the user's entry.
`is`	Causes an exact match to be found. That is, this option specifies an equality search. Use this option when you know the exact value of an user's attribute. For example, if you know the exact spelling of the user's name, use this option.
`isn't`	Returns all the entries whose attribute value does not exactly match the search string. That is, if you want to find all the users in the directory whose name is not "John Smith," use this option. Be aware, however, that use of this option can cause an extremely large number of entries to be returned to you.
`sounds like`	Causes an approximate, or phonetic, search to be performed. Use this option if you know an attribute's value, but you are unsure of the spelling. For example, if you are not sure if a user's name is spelled "Sarret," "Sarette," or "Sarett," use this option.
`starts with`	Causes a substring search to be performed. Returns all the entries whose attribute value starts with the specified search string. For example, if you know a user's name starts with "Miles," but you do not know the rest of the name, use this option.
`ends with`	Causes a substring search to be performed. Returns all the entries whose attribute value ends with the specified search string. For example, if you know a user's name ends with "Dimaggio," but you do not know the rest of the name, use this option.

- In the right-most text field, enter your search string.

To display all of the users entries contained in the Look Within directory, enter either an asterisk (*) or simply leave this text field blank.

Editing User Information

To change a user's entry, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Display the user entry as described in "Finding User Information," on page 65.
3. Edit the field corresponding to the attribute that you wish to change.

For more information, see The Edit Users Page in the online help.

NOTE It is possible that you will want to change an attribute value that is not displayed by the edit user form. In this situation, use the Directory Server `ldapmodify` command line utility, if available.

In addition, note that you can change the user's first, last, and full name field from this form, but to fully rename the entry (including the entry's distinguished name), you need to use the Rename User form. For more information on how to rename an entry, see "Renaming Users," on page 69.

Managing a User's Password

The password you set for user entries is used by the various servers for user authentication.

To change or create a user's password, perform the following steps:

1. Access the Administration Server and choose Users & Groups tab.
2. Display the user entry as described in "Finding User Information," on page 65.
3. Make the desired changes and click OK.

For more information, see The Manage Users Page in the online help.

NOTE You can change the Administration Server user from root to another user on the operating system to enable multiple users (belonging to the group) to edit/manage the configuration files. However, note that while on Unix/Linux platforms, the installer can give "rw" permissions to a group for the configuration files, on Windows NT platforms, the user must belong to the "Administrators" group.

You can also disable the user's password by clicking the Disable Password button. Doing this prevents the user from logging into a server without deleting the user's directory entry. You can allow access for the user again by using the Password Management Form to enter a new password.

Managing User Licenses

Administration Server enables you to track which iPlanet server products your users are licensed to use.

To manage the licenses available to the user, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Display the user entry as described in "Finding User Information," on page 65.
3. Click the Licenses link at the top of the User Edit form.
4. Make the desired changes and click OK.

For more information, see The Manage Users Page in the online help.

Renaming Users

The rename feature changes only the user's name; all other fields are left intact. In addition, the user's old name is still preserved so searches against the old name will still find the new entry.

When you rename a user entry, you can only change the user's name; you cannot use the rename feature to move the entry from one organizational unit to another. For example, suppose you have organizational units for Marketing and Accounting and an entry named "Billie Holiday" under the Marketing organizational unit. You can rename the entry from `Billie Holiday` to `Doc Holiday`, but you cannot rename the entry such that `Billie Holiday` under the Marketing organizational unit becomes `Billie Holiday` under the Accounting organizational unit.

To rename a user entry, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Display the user entry as described in "Finding User Information," on page 65.

   Note that if you are using common name-based DNs, specify the user's full name. If you are using uid-based distinguished names, enter the new uid value that you want to use for the entry.

3. Click the Rename User button.
4. Change the Given Name, Surname, Full Name, or UID fields as is appropriate to match the new distinguished name for the entry.
5. You can specify that the Administration Server no longer retains the old full name or `uid` values when you rename the entry by setting the `keepOldValueWhenRenaming` parameter to false. You can find this parameter in the following file:

   ```
   server_root/admin-serv/config/dsgw-orgperson.conf
   ```

For more information, see The Manage Users Page in the online help.

Removing Users

To delete a user entry, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Display the user entry as described in "Finding User Information," on page 65.
3. Click Delete User.

For more information, see The Manage Users Page in the online help.

Creating Groups

A group is an object that describes a set of objects in an LDAP database. An iPlanet Web Server group consists of users who share a common attribute. For instance, the set of objects might be a number of employees who work in the marketing division of your company. These employees might belong to a group called Marketing.

Static groups enumerate their member objects explicitly. A static group is a CN and contains `uniqueMembers` and/or `memberURLs` and/or `memberCertDescriptions`. For static groups, the members do not share a common attribute except for the `CN=<Groupname>` attribute.

For static groups, members can share a common attribute from a certificate if you use the `memberCertDescription`. Note that these will only work if the ACL uses the SSL method.

Once you create a new group, you can add users, or members, to it.

This section includes the following topics for creating groups:

- Static Groups

Static Groups

The Administration Server enables you to create a static group by specifying the same group attribute in the DNs of any number of users. A static group doesn't change unless you add a user to it or delete a user from it.

Guidelines for Creating Static Groups

Consider the following guidelines when using the Administration Server forms to create new static groups:

- Static groups can contain other static groups.
- You can optionally also add a description for the new group.
- If any organizational units have been defined for your directory, you can specify where you want the new group to be placed using the Add New Group To list. The default location is your directory's root point, or top-most entry.
- When you are finished entering the desired information, click Create Group to add the group and immediately return to the New Group form. Alternatively, click Create and Edit Group to add the group and then proceed to the Edit Group form for the group you have just added. For information on editing groups, see "Editing Group Attributes," on page 73.

To Create a Static Group

To create a static group entry, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the New Group link.
3. Enter the required information and click OK.

For more information, see The New Group Page in the online help.

"Editing Group Attributes,"page 73For more information, see The New Group Page in the online help.

Managing Groups

The Administration Server enables you to edit groups and manage group memberships from the Manage Group form. This section describes the following topics:

- Finding Group Entries

- Editing Group Attributes
- Adding Group Members
- Adding Groups to the Group Members List
- Removing Entries from the Group Members List
- Managing Owners
- Managing See Alsos
- Removing Groups
- Renaming Groups

Finding Group Entries

Before you can edit a group entry, first you must find and display the entry.

To find a group entry, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the Manage Groups link.
3. Enter the name of the group that you want to find in the Find Group field.

   You can enter any of the following values in the search field:

   - A name. Enter a full name or a partial name. All entries that equally match the search string are returned. If no such entries are found, all entries that contain the search string will be found. If no such entries are found, any entries that sounds like the search string are found.
   - An asterisk (*) to see all of the groups currently residing in your directory. You can achieve the same effect by simply leaving the field blank.
   - Any LDAP search filter. Any string that contains an equal sign (=) is considered to be a search filter.

   As an alternative, use the pull down menus in “Find all groups whose” to narrow the results of your search.
4. In the Look within field, select the organizational unit under which you want to search for entries.

   The default is the directory’s root point, or top-most entry.
5. In the Format field, choose either On-Screen or Printer.

6. Click Find.

   All the groups matching your search criteria are displayed.

7. In the resulting table, click the name of the entry that you want to edit.

The “Find all groups whose” Field

The “Find all groups whose” field allows you to build a custom search filter. Use this field to narrow down the search results that are otherwise returned by Find groups.

To display all of the group entries contained in the Look Within directory, enter either an asterisk (*) or simply leave this text field blank.

For more information regarding how to build a custom search filter, see “Building Custom Search Queries,” on page 66.

Editing Group Attributes

To edit a group entry, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the Manage Groups link.
3. Locate the group you want to edit, and type the desired changes.

   For more information regarding how to find specific entries, refer to the concepts outlined in “Finding Group Entries,” on page 72.

NOTE You can change the Administration Server user from root to another user on the operating system to enable multiple users (belonging to the group) to edit/manage the configuration files. However, note that while on Unix/Linux platforms, the installer can give “rw” permissions to a group for the configuration files, on Windows NT platforms, the user must belong to the “Administrators” group.

For more information about editing group attributes, see The Manage Groups Page in the online help.

NOTE It is possible that you will want to change an attribute value that is not displayed by the group edit form. In this situation, use the Directory Server `ldapmodify` command line utility, if available.

Adding Group Members

To add members to a group, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the Manage Groups link.
3. Locate the group you want to manage as described in “Finding Group Entries,” on page 72, and click the Edit button under Group Members.

   iPlanet Web Server displays a new form that enables you to search for entries. If you want to add user entries to the list, make sure Users is shown in the Find pull-down menu. If you want to add group entries to the group, make sure Group is shown.

4. In the right-most text field, enter a search string. Enter any of the following options:
   - A name. Enter a full name or a partial name. All entries whose name matches the search string is returned. If no such entries are found, all entries that contain the search string are found. If no such entries are found, any entries that sounds like the search string are found.
   - A user ID if you are searching for user entries.
   - A telephone number. If you enter only a partial number, any entries that have telephone numbers ending in the search number are returned.
   - An email address. any search string containing an at (@) symbol is assumed to be an email address. If an exact match cannot be found, then a search is performed to find all email addresses that begin with the search string.
   - Enter either an asterisk (*) or simply leave this text field blank to see all of the entries or groups currently residing in your directory.
   - Any LDAP search filter. Any string that contains an equal sign (=) is considered to be a search filter.
5. Click Find and Add to find all the matching entries and add them to the group.

   If the search returns any entries that you do not want add to the group, click the box in the Remove from list? column. You can also construct a search filter to match the entries you want removed and then click Find and Remove.

6. When the list of group members is complete, click Save Changes.

   The currently displayed entries are now members of the group.

For more information about adding groups members, see The Edit Members Page in the online help.

Adding Groups to the Group Members List

You can add groups (instead of individual members) to the group's members list. Doing so causes any users belonging to the included group to become a member of the receiving group. For example, if Neil Armstrong is a member of the Engineering Managers group, and you make the Engineering Managers group a member of the Engineering Personnel group, then Neil Armstrong is also a member of the Engineering Personnel group.

To add a group to the members list of another group, add the group as if it were a user entry. For more information, see "Adding Group Members," on page 74.

Removing Entries from the Group Members List

To delete an entry from the group members list, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the Manage Groups link, locate the group you want to manage as described in "Finding Group Entries," on page 72, and click the Edit button under Group Members.
3. For each member that you want to remove from the list, click the corresponding box under the Remove from list? column.

   Alternatively, you can construct a filter to find the entries you want to remove and click the Find and Remove button. For more information on creating a search filter, see "Adding Group Members," on page 74.
4. Click Save Changes. The entry(s) are deleted from the group members list.

Managing Owners

You manage a group's owners list the same way as you manage the group members list. The following table identifies which section to read for more information:

Table 4-5 Additional Information

Task You Want to Complete	Read Section
Add owners to the group	"Adding Group Members," on page 74.
Add groups to the owners list	"Adding Groups to the Group Members List," on page 75.
Remove entries from the owners list	"Removing Entries from the Group Members List," on page 75.

Managing See Alsos

"See alsos" are references to other directory entries that may be relevant to the current group. They allow users to easily find entries for people and other groups that are related to the current group.

You manage see alsos the same way as you manage the group members list. The following table shows you which section to read for more information:

Table 4-6 Additional Information

Task You Want to Complete	Read Section
Add users to see alsos	"Adding Group Members," on page 74.
Add groups to see alsos	"Adding Groups to the Group Members List," on page 75.
Remove entries from see alsos	"Removing Entries from the Group Members List," on page 75.

Removing Groups

To delete a group, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the Manage Groups link, locate the group you want to manage as described in "Finding Group Entries," on page 72, and click Delete Group.

NOTE The Administration Server does not remove the individual members of the group(s) you remove; only the group entry is removed.

Renaming Groups

To rename a group, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the Manage Groups link and locate the group you want to manage as described in "Finding Group Entries," on page 72.
3. Click the Rename Group button and type the new group name in the resulting dialog box.

When you rename a group entry, you only change the group's name; you cannot use the Rename Group feature to move the entry from one organizational unit to another. For example, a business might have the following organizations:

- organizational units for Marketing and Product Management
- a group named Online Sales under the Marketing organizational unit

In this example, you can rename the group from Online Sales to Internet Investments, but you cannot rename the entry such that Online Sales under the Marketing organizational unit becomes Online Sales under the Product Management organizational unit.

Creating Organizational Units

An organizational unit can include a number of groups, and it usually represents a division, department, or other discrete business group. A DN can exist in more than one organizational unit.

To create an organizational unit, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the New Organizational Unit link and enter the required information.

For more information, see The New Organizational Unit Page in the online help.

The following notes may be of interest to the directory administrator:

- New organizational units are created using the `organizationalUnit` object class.
- The distinguished name for new organizational units is of the form:

```
ou=new organization, ou=parent organization, ...,o=base
organization, c=country
```

For example, if you create a new organization called Accounting within the organizational unit West Coast, and your Base DN is o=Ace Industry, c=US, then the new organization unit's DN is:

```
ou=Accounting, ou=West Coast, o=Ace Industry, c=US
```

Managing Organizational Units

You edit and manage organizational units from the Organizational Unit Edit form. This section describes the following tasks:

- Finding Organizational Units
- Editing Organizational Unit Attributes
- Renaming Organizational Units
- Deleting Organizational Units

Finding Organizational Units

To find organizational units, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the Manage Organizational Units link.
3. Type the name of the unit you want to find in the Find organizational unit field. You can enter any of the following in the search field:
   - A name. Enter a full name or a partial name. All entries that equally match the search string will be returned. If no such entries are found, all entries that contain the search string will be found. If no such entries are found, any entries that sounds like the search string are found.
   - An asterisk (*) to see all of the groups currently residing in your directory. You can achieve this same result by simply leaving the field blank.
   - Any LDAP search filter. Any string that contains an equal sign (=) is considered to be a search filter.

   As an alternative, use the pull down menus in the Find all units whose field to narrow the results of your search.
4. In the Look within field, select the organizational unit under which you want to search for entries.

   The default is the root point of the directory.
5. In the Format field, choose either On-Screen or Printer.

6. Click Find.

   All the organizational units matching your search criteria are displayed.

7. In the resulting table, click the name of the organizational unit that you want to find.

The “Find all units whose” Field

The Find all units whose field allows you to build a custom search filter. Use this field to narrow down the search results that are otherwise returned by Find organizational unit.

To display all of the group entries contained in the Look Within directory, enter either an asterisk (*) or simply leave this text field blank.

For more information regarding how to build a custom search filter, see “Building Custom Search Queries,” on page 66.

Editing Organizational Unit Attributes

To change a organizational unit entry, access the Administration Server and perform the following steps:

1. Locate the organizational unit you want to edit as described in “Finding Organizational Units,” on page 78.

   The organizational unit edit form is displayed.

2. Change the displayed fields as desired and click Save Changes.

   The changes are made immediately.

NOTE It is possible that you will want to change an attribute value that is not displayed by the organizational unit edit form. In this situation, use the Directory Server `ldapmodify` command line utility, if available.

Renaming Organizational Units

To rename an organizational unit entry, access the Administration Server and perform the following steps:

1. Make sure no other entries exist in the directory under the organizational unit that you want to rename.

2. Locate the organizational unit you want to edit as described in “Finding Organizational Units,” on page 78.

3. Click the Rename button.

4. Enter the new organizational unit name in the resulting dialog box.

NOTE When you rename an organizational unit entry, you can only change the organizational unit's name; you cannot use the rename feature to move the entry from one organizational unit to another. For more information, see "Renaming Organizational Units," on page 79.

Deleting Organizational Units

To delete an organizational unit entry, access the Administration Server and perform the following steps:

1. Make sure no other entries exist in the directory under the organizational unit that you want to rename.

2. Locate the organizational unit you want to delete as described in "Finding Organizational Units," on page 78.

3. Click the Delete button.

4. Click OK in the resulting confirmation box.

   The organizational unit is immediately deleted.

Managing a Preferred Language List

iPlanet Web Server enables you to display and maintain the list of preferred languages.

To manage the preferred language list, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.

2. Click the Manage Preferred Language List link.

3. In the Display Language Selection List field, click Yes or No to specify whether iPlanet Web Server displays the Language Selection List.

4. In the Languages in the Selection List field, click the Add to List checkbox to add each language you want specified as part of the Preferred Language List.

5. Click the default value for the language you want to specify as the default language in the Preferred Language List.

6. Click Save Changes.

Chapter 5

Securing Your Web Server

This chapter describes how to activate the various security features designed to safeguard your data, deny intruders access, and allow access to those you want. iPlanet Web Server 6.0 incorporates the security architecture of all iPlanet servers: it's built on industry standards and public protocols for maximum interoperability and consistency.

Before reading this chapter you should be familiar with the basic concepts of public-key cryptography. These concepts include encryption and decryption; public and private keys; digital certificates; and the encryption protocols. For more information, see *Introduction to SSL* located at:

`http://docs.iplanet.com/docs/manuals/security/sslin/index.htm`

The process of securing your web server will be explained in detail in the following sections:

- Requiring Authentication
- Creating a Trust Database
- Requesting and Installing a VeriSign Certificate
- Requesting and Installing Other Server Certificates
- Migrating Certificates When You Upgrade
- Managing Certificates
- Installing and Managing CRLs and CKLs
- Setting Security Preferences
- Using External Encryption Modules
- Setting Client Security Requirements
- Setting Stronger Ciphers
- Considering Additional Security Issues

Requiring Authentication

Authentication is the process of confirming an identity. In the context of network interactions, authentication is the confident identification of one party by another party. Certificates are one way of supporting authentication.

Using Certificates for Authentication

A certificate consists of digital data that specifies the name of an individual, company, or other entity, and certifies that the public key, included in the certificate, belongs to that entity. Both clients and servers can have certificates.

A certificate is issued and digitally signed by a Certificate Authority, or CA. The CA can be a company that sells certificates over the Internet, or it can be a department responsible for issuing certificates for your company's intranet or extranet. You decide which CAs you trust enough to serve as verifiers of other people's identities.

In addition to a public key and the name of the entity identified by the certificate, a certificate also includes an expiration date, the name of the CA that issued the certificate, and the "digital signature" of the issuing CA. For more information regarding the content and format of a certificate, see *Introduction to SSL*.

NOTE A server certificate must be installed before encryption can be activated.

Server Authentication

Server authentication refers to the confident identification of a server by a client; that is, identification of the organization assumed to be responsible for the server at a particular network address.

Client Authentication

Client authentication refers to the confident identification of a client by a server; that is, identification of the person assumed to be using the client software. Clients can have multiple certificates, much like a person might have several different pieces of identification.

Virtual Server Certificates

You can have a different certificate database per virtual server. Each virtual server database can contain multiple certificates. Virtual servers can also have different certificates within each instance.

Creating a Trust Database

Before requesting a server certificate, you must create a trust database. In iPlanet Web Server the Administration Server and each server instance can have its own trust database. The trust database should only be created on your local machine.

When you create the trust database, you specify a password that will be used for a key-pair file. You will also need this password to start a server using encrypted communications. For a list of guidelines to consider when changing a password, see “Changing Passwords or PINs,” on page 124

In the trust database you create and store the public and private keys, referred to as your key-pair file. The key-pair file is used for SSL encryption. You will use the key-pair file when you request and install your server certificate. The certificate is stored in the trust database after installation. The key-pair file is stored encrypted in the following directory:

`server_root/``alias/<serverid-hostname>-key3.db`.

The Administration Server can only have one trust database. Each server instance can have its own trust database. Virtual servers are covered by the trust database created for their server instance.

Creating a Trust Database

To create a trust database, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click on the Create Database link.
3. Enter a password for the database.
4. Repeat.
5. Click OK.
6. For the Server Manager, click Apply, and then Restart for changes to take effect.

Using password.conf

By default, the web server prompts the administrator for the key database password before starting up. If you want to be able to restart an unattended web server, you need to save the password in a `password.conf` file. Only do this if your system is adequately protected so that this file and the key databases are not compromised.

Normally, you cannot start an Unix SSL-enabled server with the `/etc/rc.local` or the `etc/inittab` files because the server requires a password before starting. Although you can start an SSL-enabled server automatically if you keep the password in plain text in a file, this is not recommended. The server’s `password.conf` file should be owned by root or the user who installed the server, with only the owner having read and write access to them.

On Unix, leaving the SSL-enabled server's password in the `password.conf` file is a large security risk. Anyone who can access the file has access to the SSL-enabled server’s password. Consider the security risks before keeping the SSL-enabled server’s password in the `password.conf` file.

On NT, if you have an NTFS file system, you should protect the directory that contains the `password.conf` file by restricting its access, even if you do not use the file. The directory should have read/write permissions for the administration server user and the web server user. Protecting the directory prevents others from creating a false `password.conf` file. You cannot protect directories or files on FAT file systems by restricting access to them.

Start an SSL-enabled Server Automatically

If security risks are not a concern for you, follow these steps to start your SSL-enabled server automatically:

1. Make sure SSL is on.
2. Create a new `password.conf` file in the `config` subdirectory of the server instance.
   - If you are using the internal PKCS#11 software encryption module that comes with the server, enter the following information:

     `internal:`*`your_password`*

   - If you are using a different PKCS#11 module (for hardware encryption or hardware accelerators), specify the name of the PKCS#11 module, followed with the password. For example:

     `nFast:`*`your_password`*

3. Stop and restart your server for the new setting to take effect.

You will always be prompted to supply a password when starting the web server, even after the `password.conf` file has been created.

Requesting and Installing a VeriSign Certificate

VeriSign is iPlanet Web Server's preferred certificate authority. VeriSign's VICE protocol simplifies the certificate request process. VeriSign has the advantage of being able to return their certificate directly to your server.

After creating a certificate trust database for your server, you can request a certificate and submit it to a Certificate Authority (CA). If your company has its own internal CA, request your certificate from them. If you plan to purchase your certificate from a commercial CA, choose a CA and ask for the specific format of the information they require. A list of available certificate authorities including links to their sites, is available on the Request a Certificate page. For more information on what CAs may require, a list of Certificate Authorities is available through both Server Administrator, and Server Manager Security Pages under Request a Certificate.

The Administration Server can have only one server certificate. Each server instance can have its own server certificate. You can select a server instance certificate for each virtual server.

Requesting a VeriSign Certificate

To request a VeriSign Certificate, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Request VeriSign Certificate link.
3. Review the steps required.
4. Click Get Certificate.
5. Follow the VeriSign procedure.

Installing a VeriSign Certificate

If you request and receive approval for a VeriSign certificate, it should appear in the drop-down list of the Install VeriSign Certificate page in one to three days. To install a VeriSign Certificate, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Install VeriSign Certificate link.

3. Choose internal (software) from the drop-down list for cryptographic module, unless you will use an external encryption module.

4. Enter your Key Pair File Password or PIN.

5. Select the Transaction ID to Retrieve from the drop-down list.

   You will usually want the last one.

6. Click Install.

7. For the Server Manager, click Apply, and then Restart for changes to take effect.

Requesting and Installing Other Server Certificates

Besides VeriSign, you can request and install certificates from other certificate authorities. A list of CAs is available through both Server Administrator, and Server Manager Security Pages under Request a Certificate. Your company or organization may provide its own internal certificates. This section describes how you would request and install these other types of server certificates.

Required CA Information

Before you begin the request process, make sure you know what information your CA requires. Whether you are requesting a server certificate from a commercial CA or an internal CA, you need to provide the following information:

- **Common Name** must be the fully qualified hostname used in DNS lookups (for example, *www.iplanet.com*). This is the hostname in the URL that a browser uses to connect to your site. If these two names don't match, a client is notified that the certificate name doesn't match the site name, creating doubt about the authenticity of your certificate. Some CAs might have different requirements, so it's important to check with them.

  You can also enter wildcard and regular expressions in this field if you are requesting a certificate from an internal CA. Most vendors would not approve a certificate request with a wildcard or regular expression entered for common name.

- **Email Address** is your business email address. This is used for correspondence between you and the CA.
- **Organization** is the official, legal name of your company, educational institution, partnership, and so on. Most CAs require that you verify this information with legal documents (such as a copy of a business license).
- **Organizational Unit** is an optional field that describes an organization within your company. This can also be used to note a less formal company name (without the *Inc.*, *Corp.*, and so on).
- **Locality** is an optional field that usually describes the city, principality, or country for the organization.
- **State or Province** is usually required, but can be optional for some CAs. Note that most CAs won't accept abbreviations, but check with them to be sure.
- **Country** is a required, two-character abbreviation of your country name (in ISO format). The country code for the United States is US.

All this information is combined as a series of attribute-value pairs called the distinguished name (DN), which uniquely identifies the subject of the certificate.

If you are purchasing your certificate from a commercial CA, you must contact the CA to find out what additional information they require before they issue a certificate. Most CAs require that you prove your identity. For example, they want to verify your company name and who is authorized by the company to administer the server, and they might ask whether you have the legal right to use the information you provide.

Some commercial CAs offer certificates with greater detail and veracity to organizations or individuals who provide more thorough identification. For example, you might be able to purchase a certificate stating that the CA has not only verified that you are the rightful administrator of the www.iplanet.com computer, but that you are a company that has been in business for three years, and have no outstanding customer litigation.

Requesting Other Server Certificates

To request a certificate, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Request a Certificate link.

3. Select if this is a new certificate or a certificate renewal.

   Many certificates expire after a set period of time, such as six months or a year. Some CAs will automatically send you a renewal.

4. Perform the following steps to specify how you want to submit the request for the certificate:

   ❍ If the CA expects to receive the request in an email message, check CA Email and enter the email address of the CA. For a list of CAs, click List of available certificate authorities.

   ❍ If you are requesting the certificate from an internal CA that is using Netscape Certificate Server, click CA URL and enter the URL for the Certificate Server. This URL should point to the certificate server's program that handles certificate requests. A sample URL might be: `https://CA.mozilla.com:444/cms`.

5. Select the cryptographic module for the key-pair file you want to use when requesting the certificate from the drop-down list.

6. Enter the password for your key-pair file.

   This is the password you specified when you created the trust database, unless you selected a cryptographic module other than the internal module. The server uses the password to get your private key and encrypt a message to the CA. The server then sends both your *public key* and the encrypted message to the CA. The CA uses the public key to decrypt your message.

7. Enter your identification information.

   The format of this information varies by CA. For a general description of these fields, alist of Certificate Authorities is available through both Server Administrator, and Server Manager Security Pages under Request a Certificate. Note that most of this information usually isn't required for a certificate renewal.

8. Double-check your work to ensure accuracy.

   The more accurate the information, the faster your certificate is likely to be approved. If your request is going to a certificate server, you'll be prompted to verify the form information before the request is submitted.

9. Click OK.

10. For the Server Manager, click Apply, and then Restart for changes to take effect.

The server generates a certificate request that contains your information. The request has a digital signature created with your private key. The CA uses a digital signature to verify that the request wasn't tampered with during routing from your server machine to the CA. In the rare event that the request is tampered with, the CA will usually contact you by phone.

If you choose to email the request, the server composes an email message containing the request and sends the message to the CA. Typically, the certificate is then returned to you via email. If instead you specified a URL to a certificate server, your server uses the URL to submit the request to the Certificate Server. You might get a response via email or other means depending on the CA.

The CA will notify you if it agrees to issue you a certificate. In most cases, the CA will send your certificate via email. If your organization is using a certificate server, you may be able to search for the certificate by using the certificate server's forms.

NOTE Not everyone who requests a certificate from a commercial CA is given one. Many CAs require you to prove your identity before issuing you a certificate. Also, it can take anywhere from one day to two months to get approval. You are responsible for promptly providing all the necessary information to the CA.

Once you receive the certificate, you can install it. In the meantime, you can still use your server without SSL.

Installing Other Server Certificates

When you receive your certificate back from the CA, it will be encrypted with your public key so that only you can decrypt it. Only by entering the correct password for your trust database, can you decrypt and install your certificate.

There are three types of certificates:

- Your own server's certificate to present to clients
- A CA's own certificate for use in a certificate chain
- A trusted CA's certificate

A certificate chain is a hierarchical series of certificates signed by successive certificate authorities. A CA certificate identifies a certificate authority (CA) and is used to sign certificates issued by that authority. A CA certificate can in turn be signed by the CA certificate of a parent CA, and so on, up to a root CA.

NOTE If your CA doesn't automatically send you their certificate, you should request it. Many CAs include their certificate in the email with your certificate, and your server installs both certificates at the same time.

When you receive a certificate from the CA, it will be encrypted with your public key so that only you can decrypt it. The server will use the key-pair file password you specify to decrypt the certificate when you install it. You can either save the email somewhere accessible to the server, or copy the text of the email and be ready to paste the text into the Install Certificate form, as described here.

Installing a Certificate

To install a certificate, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Install Certificate link.

3. Check the type of certificate you are installing:
   - This Server is for a single certificate associated only with your server.
   - Server Certificate Chain is for a CA's certificate to include in a certificate chain.
   - Trusted Certificate Authority (CA) is for a certificate of a CA that you want to accept as a trusted CA for client authentication.
4. Select the Cryptographic Module from the drop-down list.
5. Enter the Key-Pair File Password.
6. Leave the a name for the certificate field blank if it will be the only one used for this server instance, unless:
   - Multiple certificates will be used for virtual servers

     Enter a certificate name unique within the server instance
   - Cryptographic modules other than internal are used

     Enter a certificate name unique across all server instances within a single cryptographic module

   If a name is entered, it will be displayed in the Manage Certificates list, and should be descriptive. For example, "United States Postal Service CA" is the name of a CA, and "VeriSign Class 2 Primary CA" describes both a CA and the type of certificate. When no certificate name is entered, the default value is applied.
7. Select either:
   - Message is in this file and enter the full pathname to the saved email
   - Message text (with headers) and paste the email text

     If you copy and paste the text, be sure to include the headers "Begin Certificate" and "End Certificate"—including the beginning and ending hyphens.
8. Click OK.
9. Select either:
   - Add Certificate if you are installing a new certificate.
   - Replace Certificate if you are installing a certificate renewal.
10. For the Server Manager, click Apply, and then Restart for changes to take effect.

The certificate is stored in the server's certificate database. The filename will be `<alias>-cert7.db`. For example:

```
https-serverid-hostname-cert7.db
```

Migrating Certificates When You Upgrade

If you are upgrading from iPlanet Web Server 4.x, your files, including your trust and certificate databases, will be updated automatically.

If you are upgrading from an Enterprise Server 3.x, you will need to migrate your trust and certificate databases. Make sure that iPlanet Web Server 6.0 Administration Server user has read and write permissions on the old 3.x database files. The files are `<alias>-cert.db` and `<alias>-key.db`, located in the `<3.x_server_root>/alias` directory.

Key-pair files and certificates are migrated only if your server has security enabled. You can also migrate keys and certificates by themselves using the Security tabs in the Administration Server page and the Server Manager page.

In previous versions, a certificate and key-pair file was referred to by an alias which could be used by multiple server instances. The Administration Server managed all the aliases and their constituent certificates. In iPlanet Web Server 6.0, the Administration Server and each server instance has its own certificate and key-pair file, referred to as a trust database instead of an alias.

You manage the trust database and its constituent certificates, including the server certificate and all the included Certificate Authorities, from the Administration Server for its self, and from the Server Manager for server instances. The certificate and key-pair database files are now named after the server instance that uses them. If in the previous version, multiple server instances shared the same alias, when migrated the certificate and key-pair file are renamed for the new server instance.

The entire trust database associated with the server instance is migrated. All the Certificate Authorities listed in your previous database are migrated to the iPlanet Web Server 6.0 database. If duplicate CAs occur, use the previous CA until it expires. Do not attempt to delete duplicate CAs.

Migrating a Certificate

To migrate a certificate, perform the following steps:

1. From your local machine, access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Choose:

   - Migrate 3.X Certificates link from the Administration Server
   - Migrate Certificate link from the Server Manager.

3. Enter the 3.6 Server Root.
4. Enter the Alias.
5. Enter the Password.
6. Click OK.
7. For the Server Manager, click Apply, and then Restart for changes to take effect.

Using the Built-in Root Certificate Module

The dynamically loadable root certificate module included with iPlanet Web Server 6.0 contains the root certificates for many CAs, including VeriSign. The root certificate module allows you to upgrade your root certificates to newer versions in a much easier way than before. In the past, you were required to delete the old root certificates one at a time, then install the new ones one at a time. To install well-known CA certificates, you can now simply update the root certificate module file to a newer version as it becomes available through future versions of iPlanet Web Server, or in Service Packs.

Because the root certificate is implemented as a PKCS#11 cryptographic module, you can never delete the root certificates it contains, and the option to delete will not be offered when managing these certificates. To remove the root certificates from your server instances, you can disable the root certificate module by deleting the following in the server's `alias` file:

- `libnssckbi.so` (on most Unix platforms)
- `libnssckbi.sl` (on HP-UX)
- `nssckbi.dll` (on NT)

If you later wish to restore the root certificate module, you can copy the extension from `bin/https/lib` (Unix and HP) or `bin\https\bin` (NT) back into the `alias` subdirectory.

You can modify the trust information of the root certificates. The trust information is written to the certificate database for the server instance being edited, not back to the root certificate module itself.

Managing Certificates

You can view, delete, or edit the trust settings of the various certificates installed on your server. This includes your own certificate and certificates from CAs.

To manage certificate lists, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Manage Certificates link.

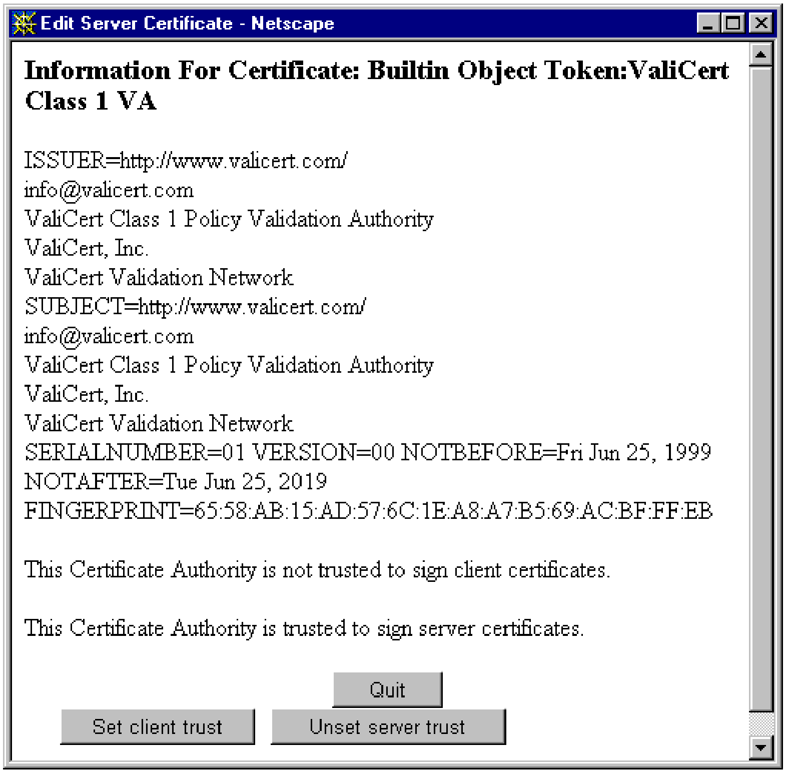
   - If you are managing a certificate for a default configuration using the internal cryptographic module, a list of all installed certificates with their type and expiration date is displayed. All certificates are stored in the directory *`server_root/`*`alias.`

   - If you are using an external cryptographic module, such as a hardware accelerator, you will first need to enter your password for each specific module and click OK. The certificate list will update to include certificates in the module.

3. Click the Certificate Name you wish to manage.

   An Edit Server Certificate page appears with management options for that type of certificate. Only CA certificates will allow you to set or unset client trust. Some external cryptographic modules will not allow certificates to be deleted.

Figure 5-1 Edit Server Certificate



Information For Certificate: Builtin Object Token:ValiCert Class 1 VA

ISSUER=http://www.valicert.com/
info@valicert.com
ValiCert Class 1 Policy Validation Authority
ValiCert, Inc.
ValiCert Validation Network
SUBJECT=http://www.valicert.com/
info@valicert.com
ValiCert Class 1 Policy Validation Authority
ValiCert, Inc.
ValiCert Validation Network
SERIALNUMBER=01 VERSION=00 NOTBEFORE=Fri Jun 25, 1999
NOTAFTER=Tue Jun 25, 2019
FINGERPRINT=65:58:AB:15:AD:57:6C:1E:A8:A7:B5:69:AC:BF:FF:EB

This Certificate Authority is not trusted to sign client certificates.

This Certificate Authority is trusted to sign server certificates.

4. In the Edit Server Certificate window you may select:

- Delete Certificate or Quit for certificates obtained internally
- Set client trust, Unset server trust, or Quit for CA certificates

5. Click OK.

6. For the Server Manager, click Apply, and then Restart for changes to take effect.

Certificate information includes the owner and who issued it.

Trust settings allow you to set client trust or unset server trust. For LDAP server certificates the server must be trusted.

Installing and Managing CRLs and CKLs

Certificate revocation lists (CRLs) and compromised key lists (CKLs) make known any certificates and keys that either client or server users should no longer trust. If data in a certificate changes, for example, a user changes offices or leaves the organization before the certificate expires, the certificate is revoked, and its data appears in a CRL. If a key is tampered with or otherwise compromised, the key and its data appear in a CKL. Both CRLs and CKLs are produced and periodically updated by a CA.

Installing a CRL or CKL

To obtain a CRL or CKL from a CA, perform the following steps:

1. Obtain the CA’s URL for downloading CRLs or CKLs.
2. Enter the URL in your browser to access the site.
3. Follow the CA's instructions for downloading the CRL or CKL to a local directory.
4. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.
5. Click the Install CRL/CKLs link.
6. Select either:
   - Certificate Revocation List
   - Compromised Key List
7. Enter the full path name to the associated file.
8. Click OK.
   - If you selected Certificate Revocation List, the Add Certificate Revocation List page will appear listing CRL information.
   - If you selected Compromised Key List, the Add Compromised Key List page will appear listing CKL information.

NOTE If a CRL or CKL list already exists in the database, a Replace Certificate Revocation List or Replace Compromised Key List page will appear.

9. Click Add.

10. Click OK.

11. For the Server Manager, click Apply, and then Restart for changes to take effect.

Managing CRLs and CKLs

To manage CRLs and CKLs, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Manage CRL/CKLs link.

   The Manage Certificate Revocation Lists /Compromised Key Lists page appears with all installed Server CRLs and CKLs listed along with their expiration dates.

3. Select a Certificate Name from either the Server CRLs or Server CKLs list.

4. Choose:

   - Delete CRL
   - Delete CKL

5. For the Server Manager, click Apply, and then Restart for changes to take effect.

Setting Security Preferences

Once you have a certificate, you can begin securing your server. Several security elements are provided by iPlanet Web Server.

Encryption is the process of transforming information so it is unintelligible to anyone but the intended recipient. Decryption is the process of transforming encrypted information so that it is intelligible again. iPlanet Web Server 6.0 includes supports SSL and TLS encryption protocols.

A cipher is a cryptographic algorithm (a mathematical function), used for encryption or decryption. SSL and TLS protocols contain numerous cipher suites. Some ciphers are stronger and more secure than others. Generally speaking, the more bits a cipher uses, the harder it is to decrypt the data.

In any two-way encryption process, both parties must use the same ciphers. Because a number of ciphers are available, you need to enable your server for those most commonly used.

During a secure connection, the client and the server agree to use the strongest cipher they can both have for communication. You can choose ciphers from the SSL2, SSL3, and TLS protocols.

NOTE Improvements to security and performance were made after SSL version 2.0; you should not use SSL 2 unless you have clients that are not capable of using SSL 3. Client certificates are not guaranteed to work with SSL 2 ciphers.

The encryption process alone isn't enough to secure your server's confidential information. A key must be used with the encrypting cipher to produce the actual encrypted result, or to decrypt previously encrypted information. The encryption process uses two keys to achieve this result: a public key and a private key. Information encrypted with a public key can be decrypted only with the associated private key. The public key is published as part of a certificate; only the associated private key is safeguarded.

For description of the various cipher suites, and more information about keys and certificates, see *Introduction to SSL*.

To specify which ciphers your server can use, check them in the list. Unless you have a compelling reason not to use a specific cipher, you should check them all. However, you may not wish to enabling ciphers with less than optimal encryption.

CAUTION Do not select "No Encryption, only MD5 message authentication". If no other ciphers are available on the client side, the server will default to this setting and no encryption will occur.

SSL and TLS Protocols

iPlanet Web Server 6.0 supports the Secure Sockets Layer (SSL) and the Transport Layer Security (TLS) protocols for encrypted communication. SSL and TLS are application independent, and higher level protocols can be layered transparently on them.

SSL and TLS protocols support a variety of ciphers used to authenticate the server and client to each other, transmit certificates, and establish session keys. Clients and servers may support different cipher suites, or sets of ciphers, depending on factors such as which protocol they support, company policies on encryption strength, and government restrictions on export of encrypted software. Among other functions, the SSL and TLS handshake protocols determine how the server and client negotiate which cipher suites they will use to communicate.

Using SSL to Communicate with LDAP

You should require your Administration Server to communicate with LDAP using SSL. To enable SSL on your Administration Server, perform the following steps:

1. Access the Administration Server and choose the Global Settings tab.
2. Click the Configure Directory Service link.
3. Select Yes to use Secure Sockets Layer (SSL) for connections.
4. Click Save Changes.
5. Click OK to change your port to the standard port for LDAP over SSL.

Enabling Security for Connection Groups

You can secure your server’s connection groups by:

- Turning the security on
- Selecting a server certificate for a connection group
- Selecting ciphers

Turning Security On

You must turn security on before you can configure the other security settings for your connection group. You can turn security on when you create a new listen socket, or when you edit an existing listen socket.

Turning Security On When Creating a Listen Socket

To turn security on when creating a new listen socket, perform the following steps:

1. Access the Server Manager and select the server instance the listen socket will be created in from the drop-down list.
2. Select the Preferences tab, if not already displayed.
3. Choose the Add Listen Socket link.

   The Create a Listen Socket page is displayed.
4. Enter the required information and select a default virtual server.
5. Turn Security on using the drop-down list.
6. Click OK
7. Click Apply, and then Restart for changes to take effect.

NOTE You will need to use the Edit Listen Sockets link to configure the security settings after a listen socket is created.

Turning Security On When Editing a Listen Socket

You can also turn security on when editing a listen socket from either the Administration Server or the Server Manager. To turn security on when editing a listen socket, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.
2. Select the Preferences tab, if not already displayed.
3. Choose the Edit Listen Sockets link.

   The Listen Sockets Table page is displayed.
4. Use the drop-down Action list to select Edit, if not already displayed, for the connection group you want to secure.
5. Use the drop-down list in the Security column to turn security on for the connection group.

6. Click OK.

   The Attributes link will now be displayed in the Security column.

7. For the Server Manager, click Apply, and then Restart for changes to take effect.

Selecting a Server Certificate for a Connection Group

You can configure connection groups in either the Administration Server or the Server Manager to use server certificates you have requested and installed.

NOTE You must have at least one certificate is installed.

To select a server certificate for your connection group to use, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Preferences tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Edit Listen Sockets link.

   The Listen Socket Table page appears.

3. Use the drop-down Action list to select Edit, if not already displayed, for the connection group you are selecting a certificate for.

4. Use the drop-down list to turn Security on for that connection group, if it is off.

5. Click the Attributes link.

   The Security Settings of Listen Socket page appears.

NOTE If you have an external module installed, the Manage Server Certificates page will appear requiring the external module’s password before you can continue.

6. Select a server certificate from the drop-down CertificateName list for the connection group.

   The list contains all internal and external certificates installed.

7. Click OK

8. For the Server Manager, click Apply, and then Restart for changes to take effect.

Selecting Ciphers

To protect the security of your web server, you should enable SSL. You can enable the SSL 2.0, SSL 3.0, and TLS encryption protocols and select the various cipher suites. SSL and TLS can be enabled on the connection group for the Administration Server. Enabling SSL and TLS on a connection group for the Server Manager will set those security preferences for all virtual servers associated with that connection group.

If you wish to have unsecured virtual servers, they must all be configured to the same connection group with security turned off.

The default settings allow the most commonly used ciphers. Unless you have a compelling reason why you don’t want to use a specific cipher suite, you should allow them all. For more information regarding specific ciphers, see *Introduction to SSL*

NOTE You must have at least one certificate installed.

To enable SSL and TLS, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Preferences tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Edit Listen Sockets link.

   The Listen Socket Table page appears.

3. Use the drop-down Action list to select Edit, if not already displayed, for the connection group you are enabling security for.

4. Use the drop-down list to turn Security on for that connection group, if it is off.

5. Click OK.

   The Attributes link now appears.

6. Click the Attributes link.

   The Security Settings of Listen Socket page appears.

NOTE If you have an external module installed, the Manage Server Certificates page will appear requiring the external module’s password before you can continue.

7. Select either:
   - Cipher Default
   - SSL2
   - SSL3 /TLS
8. (Optional) If you selected SSL2 or SSL3/TLS, in the Security Features window either:
   - Accept Allow and the default ciphers
   - Accept Allow and check only desired ciphers, or uncheck unwanted ciphers
   - Uncheck Allow to disable this protocol and all its ciphers

NOTE Check both TLS and SSL3 for Netscape Navigator 6.0. Use the TLS Rollback option for Microsoft Internet Explorer 5.0 and 5.5. TLS must also be enabled on the browser seeking access to your server. For TLS Rollback also check TLS, and make sure both SSL3 and SSL2 are disabled.

9. Click OK to close the Security Features window.
10. Click OK
11. For the Server Manager, click Apply, and then Restart for changes to take effect.

NOTE When you apply changes after turning on security for a connection group, the `magnus.conf` file is automatically modified to show security on, and all virtual servers associated with the connection group are automatically assigned the default security parameters.

Once you have enabled SSL on a server, its URLs use `https` instead of `http`. URLs that point to documents on an SSL-enabled server have this format:

`https://`*`servername.`*`[`*`domain.`*`[`*`dom`*`]]`*`:`*`[`*`port#`*`]`

For example, `https://admin.iplanet.com:443`.

If you use the default secure http port number (443), you don’t have to enter the port number in the URL.

Configuring Security Globally

Installing an SSL-enabled server creates directive entries in the `magnus.conf` file (the server's main configuration file) for global security parameters. Security must be set to 'on' for virtual server security settings to work. SSL properties for virtual servers can be found on a per-server basis in the `SSLPARAMS` element of the `server.xml` file.

To set values for your SSL configuration file directives, perform the following steps:

1. Access the Server Manager and select the server instance of the virtual server from the drop-down list.
2. Select the Preferences tab, if not already selected.
3. Choose the Edit Listen Sockets link.
4. Turn Security On for the listen socket you will set values for, if it isn't already on.
5. Click OK.
6. Go to the Magnus Editor link.
7. Select SSL Settings from the drop-down list and click Manage.
8. Enter the values for:
   - SSLSessionTimeout
   - SSLCacheEntires
   - SSL3SessionTimeout
9. Click OK
10. Click Apply, and then Restart for changes to take effect.

These SSL Configuration File Directives are described below:

SSLSessionTimeout

The `SSLSessionTimeout` directive controls SSL2 session caching.

Syntax

`SSLSessionTimeout` seconds

`seconds` is the number of seconds until a cached SSL session becomes invalid. The default value is 100. If the `SSLSessionTimeout` directive is specified, the value of seconds is silently constrained to be between 5 and 100 seconds.

SSLCacheEntries

Specifies the number of SSL sessions that can be cached.

SSL3SessionTimeout

The SSL3SessionTimeout directive controls SSL3 and TLS session caching.

Syntax

`SSL3SessionTimeout` seconds

`seconds` is the number of seconds until a cached SSL3 session becomes invalid. The default value is 86400 (24 hours). If the `SSL3SessionTimeout` directive is specified, the value of seconds is silently constrained to be between 5 and 86400 seconds.

NOTE A single connection group on a listen socket must have the same SSLPARAMS; multiple groups can have different SSLPARAMS.

Using External Encryption Modules

iPlanet Web Server 6.0 supports the following methods of using external cryptographic modules such as smart cards or token rings:

- PKCS#11
- FIPS-140

You will need to add the PKCS #11 module before activating the FIPS-140 encryption standard.

Installing the PKCS#11Module

iPlanet Web Server supports Public Key Cryptography Standard (PKCS) #11, which defines the interface used for communication between SSL and PKCS#11 modules. PKCS#11 modules are used for standards-based connectivity to SSL hardware accelerators. Imported certificates and keys for external hardware accelerators are stored in the `secmod.db` file, which is generated when the PKCs#11 module is installed.

Using modutil to Install a PKCS#11 Module

You can install PKCS#11 modules in the form of `.jar` files or object files using the `modutil` tool.

To install the PKCS#11 module using `modutil`, perform the following steps:

1. Make sure all servers, including the Administration server, are turned off.
2. Go to the `server_root/alias directory` containing the databases.
3. Add `server_root/bin/https/admin/bin` to your PATH.
4. Locate `modutil` in `server_root/bin/https/admin/bin`.
5. Set the environment. For example:
   - On Unix: `setenv`

     `LD_LIBRARY_PATH server_root/bin/https/lib:${LD_LIBRARY_PATH}`
   - On IBM-AIX: `LIBPATH`
   - On HP-UX: `SHLIB_PATH`
   - On NT, add it to the `PATH`

     `LD_LIBRARY_PATH server_root/bin/https/bin`

     You can find the PATH for your machine listed under: `server_root/https-admin/start`.
6. Enter the command: `modutil`.

   The options will be listed.
7. Perform the actions required.

   For example, to add the PCKS#11 module in Unix you would enter:

   `modutil -add` (the name of PCKS#11 file) `-libfile` (your `libfile` for PCKS#11) `-nocertdb -dbdir .` (your db directory)

Using pk12util

The `pk12util` allows you to export certificates and keys from your internal database and to import them into an internal or external PKCS#11 module. You can always export certificates and keys to your internal database, but most external tokens will not allow you to export certificates and keys. By default, `pk12util` uses certificate and key databases named `cert7.db` and `key3.db`.

Exporting with pk12util

To export a certificate and key from an internal database, perform the following steps:

1. Go to the `server_root/alias directory` containing the databases.
2. Add `server_root/bin/https/admin/bin` to your PATH.
3. Locate `pk12util` in `server_root/bin/https/admin/bin`.
4. Set the environment. For example:
   - On Unix: `setenv`

     `LD_LIBRARY_PATH/server_root/bin/https/lib:${LD_LIBRARY_PATH}`
   - On IBM-AIX: `LIBPATH`
   - On HP-UX: `SHLIB_PATH`
   - On NT, add it to the `PATH`

     `LD_LIBRARY_PATH server_root/bin/https/bin`

     You can find the PATH for your machine listed under: `server_root/https-admin/start`.
5. Enter the command: `pk12util`.

   The options will be listed.
6. Perform the actions required.

   For example, in Unix you would enter:

   ```
   pk12util -o certpk12 -n Server-Cert [-d /server/alias] [-P
   https-test-host]
   ```
7. Enter the database password.
8. Enter `pkcs12` password.

Importing with pk12util

To import a certificate and key into an internal or external PKCS#11 module, perform the following steps:

1. Go to the `server_root/alias directory` containing the databases.
2. Add `server_root/bin/https/admin/bin` to your PATH.
3. Locate `pk12util` in `server_root/bin/https/admin/bin`.

4. Set the environment. For example:

   - On Unix: `setenv`

     `LD_LIBRARY_PATH/server_root/bin/https/lib:${LD_LIBRARY_PATH}`

   - On IBM-AIX: `LIBPATH`

   - On HP-UX: `SHLIB_PATH`

   - On NT, add it to the `PATH`

     `LD_LIBRARY_PATH server_root/bin/https/bin`

     You can find the PATH for your machine listed under: `server_root/https-admin/start`.

5. Enter the command: `pk12util`.

   The options will be listed.

6. Perform the actions required.

   For example, in Unix you would enter:

   ```
   pk12util -i pk12_sunspot [-d certdir][-h “nCipher”][-P
   https-jones.redplanet.com-jones-]
   ```

   -P must follow the -h and be the last argument.

   Enter the exact token name including capital letters and spaces between quote marks.

7. Enter the database password.

8. Enter `pkcs12` password.Starting the Server with an External Certificate

If you install a certificate for your server into an external PKCS#11 module (for example, a hardware accelerator), the server will not be able to start using that certificate until you edit the `server.xml`, or specify the certificate name as described below.

The server always tries to start with the certificate named “Server-Cert.” However, certificates in external PKCS#11 modules include one of the module’s token names in their identifier. For example, a server certificate installed on an external smartcard reader called “smartcard0” would be named “smartcard0:Server-Cert.”

To start a server with a certificate installed in an external module, you’ll need to specify the certificate name for the connection group it runs on.

Selecting the Certificate Name for a Connection Group

To select the certificate name for the connection group, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Preferences tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Select the Preferences tab, if not already selected.
3. Click the Edit Listen Sockets link.

   The Listen Socket Table page appears.

4. Use the drop-down Action list to select Edit, if not already displayed, for the connection group you are enabling security for.
5. Use the drop-down list to turn Security on for that connection group, if it is off.
6. Click OK.

   The Attributes link now appears.

7. Click the Attributes link.

   The Security Settings of Listen Socket page appears.

8. Use the drop-down CertificateName list to select the external server certificate.
9. Click OK
10. For the Server Manager, click Apply, and then Restart for changes to take effect.

You could also tell the server to start with that server certificate instead, by manually editing the `server.xml` file. Change the `servercertnickname` attribute in the SSLPARAMS to:

```
$TOKENNAME:Server-Cert
```

To find what value to use for `$TOKENNAME`, go to the server's Security tab and select the Manage Certificates link. When you log in to the external module where Server-Cert is stored, its certificates are displayed in the list in the `$TOKENNAME:$NICKNAME` form.

NOTE If you did not create a trust database, one will be created for you when you request or install a certificate for an external PKCS#11 module. The default database created has no password and cannot be accessed. Your external module will work, but you will not be able to request and install server certificates. If a default database has been created without a password, use the Security tab Create Database page to set the password.

FIPS-140 Standard

PKCS#11 APIs enable communication with software or hardware modules that perform cryptographic operations. Once PKCS#11 is installed on your server, you can configure iPlanet Web Server to be Federal Information Processing Standards (FIPS)-140 compliant. These libraries are included only in SSL version 3.0.

To enable FIPS-140, perform the following steps:

1. Install the plug-in following the FIPS-140 instructions.
2. Access either the Administration Server or the Server Manager and choose the Preferences tab.

   For the Server Manager you must first select the server instance from the drop-down list.
3. Click the Edit Listen Sockets link.

   The Listen Socket Table page appears.
4. Use the drop-down Action list to select Edit, if not already displayed, for the connection group you are enabling FIPS-140 on.
5. Use the drop-down list to turn Security on for that connection group, if it is off.
6. Click OK.

   The Attributes link now appears.
7. Click the Attributes link.
8. The Security Settings of Listen Socket page appears.
9. Click the SSL3/TLS link.

   The Security Feature window appears.
10. Check Allow: SSL version 3, if it is not already checked.
11. Select the appropriate FIPS-140 cipher suite:
    - (FIPS) DES with 56 bit encryption and SHA message authentication
    - (FIPS) Triple DES with 168 bit encryption and SHA message authentication
12. Click OK to close the Security Features window.
13. Click OK
14. For the Server Manager, click Apply, and then Restart for changes to take effect.

Setting Client Security Requirements

After you have performed all of the steps to secure your servers, you can set additional security requirements for your clients.

Requiring Client Authentication

You can enable the connection groups for your Administration Server and each server instance to require client authentication. When client authentication is enabled, the client's certificate is required before the server will send a response to a query.

iPlanet Web Server supports authenticating client certificates by matching the CA in the client certificate with a CA trusted for signing client certificates. You can view a list of CAs trusted for signing client certificates in the Manage Certificates page under Security in the Administration Server. There are four types of CAs:

- Untrusted CA (will not be matched)
- Trusted Server CA (will not be matched)
- Trusted Client CA (will be matched)
- Trusted Client/Server CA (will be matched)

You can configure the web server to refuse any client that doesn't have a client certificate from a trusted CA. To accept or reject trusted CAs, you must have set client trust for the CA. For more information, see "Managing Certificates," on page 96.

iPlanet Web Server will log an error, reject the certificate, and return a message to the client if the certificate has expired. You can also view which certificates have expired in the Administration Servers Manage Certificates page.

You can configure your server to gather information from the client certificate and match it with a user entry in an LDAP directory. This ensures that the client has a valid certificate and an entry in the LDAP directory. It can also ensure that the client certificate matches the one in the LDAP directory. To learn how to do this, see "Mapping Client Certificates to LDAP," on page 114.

You can combine client certificates with access control, so that in addition to being from a trusted CA, the user associated with the certificate must match the access control rules (ACLs). For more information, see "Using Access Control Files," on page 161.

You can also process information from client certificates. For more information, see the *NSAPI Programmer's Guide.*

To Require Client Authentication

To require client authentication, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Preferences tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Edit Listen Sockets link.

   The Listen Socket Table page appears.

3. Use the drop-down Action list to select Edit, if not already displayed, for the connection group you are requiring client authentication for.

4. Use the drop-down list to turn Security on for that connection group, if it is off.

5. Click the Attributes link.

   The Security Settings of Listen Socket page appears.

6. Click Off for Client Auth to turn it on.

7. Click OK.

8. For the Server Manager, click Apply, and then Restart for changes to take effect.

NOTE Currently, there is a single certificate trust database per web server instance. All the secure virtual servers running under that server instance share the same list of trusted client CAs. If two virtual servers require different trusted CAs, then these virtual servers should be run in different server instances with separate trust databases.

Mapping Client Certificates to LDAP

This section describes the process iPlanet Web Server uses to map a client certificate to an entry in an LDAP directory.

When the server gets a request from a client, it asks for the client's certificate before proceeding. Some clients send the client certificate to the server along with the request.

NOTE Before mapping client certificates to LDAP, you also need to set up the required ACLs; for more information, see Chapter 8, "Controlling Access to Your Server."

The server tries to match the CA to the list of trusted CAs in the Administration Server. If there isn't a match, iPlanet Web Server ends the connection. If there is a match, the server continues processing the request.

After verifying the certificate is from a trusted CA, the server maps the certificate to an LDAP entry by:

- Mapping the issuer and subject DN from the client certificate to a branch point in the LDAP directory.
- Searching the LDAP directory for an entry that matches the information about the subject (end-user) of the client certificate.
- (Optional) Verifying the client certificate with one in the LDAP entry that corresponds to the DN.

The server uses a certificate mapping file called `certmap.conf` to determine how to do the LDAP search. The mapping file tells the server what values to take from the client certificate (such as the end-user's name, email address, and so on). The server uses these values to search for a user entry in the LDAP directory, but first the server needs to determine where in the LDAP directory it needs to start its search. The certificate mapping file also tells the server where to start.

Once the server knows where to start its search and what it needs to search for (step 1), it performs the search in the LDAP directory (step 2). If it finds no matching entry or more than one matching entry, and the mapping is *not* set to verify the certificate, the search fails. For a complete list of the expected search result behavior, see the following Table 5-1 table. Note that you can specify the expected behavior in the ACL; for example, you can specify that iPlanet Web Server accepts only you if the certificate match fails. For more information regarding how to set the ACL preferences, see "Using Access Control Files," on page 161.

Table 5-1 LDAP Search Results

LDAP Search Result	Certificate Verification ON	Certificate Verification OFF
No entry found	Authentication fails	Authentication fails
Exactly one entry found	Authentication fails	Authentication succeeds
More than one entry found	Authentication fails	Authorization fails

After the server finds a matching entry and certificate in the LDAP directory, it can use that information to process the transaction. For example, some servers use certificate-to-LDAP mapping to determine access to a server.

Using the certmap.conf File

Certificate mapping determines how a server looks up a user entry in the LDAP directory. You can use `certmap.conf` to configure how a certificate, designated by name, is mapped to an LDAP entry. You edit this file and add entries to match the organization of your LDAP directory and to list the certificates you want your users to have. Users can be authenticated based on userid, email, or any other value used in the `subjectDN`. Specifically, the mapping file defines the following information:

- Where in the LDAP tree the server should begin its search
- What certificate attributes the server should use as search criteria when searching for the entry in the LDAP directory
- Whether or not the server goes through an additional verification process

The certificate mapping file is located in the following location:

```
server_root/userdb/certmap.conf
```

The file contains one or more named mappings, each applying to a different CA. A mapping has the following syntax:

```
certmap <name> <issuerDN>

<name>:<property> [<value>]
```

The first line specifies a name for the entry and the attributes that form the distinguished name found in the CA certificate. The name is arbitrary; you can define it to be whatever you want. However, `issuerDN` must exactly match the issuer DN of the CA who issued the client certificate. For example, the following two `issuerDN` lines differ only in the spaces separating the attributes, but the server treats these two entries as different:

```
certmap iplanet1 ou=iPlanet Certificate Authority,o=iPlanet,c=US
certmap iplanet2 ou=iPlanet Certificate Authority,o=iPlanet, c=US
```

TIP If you are using iPlanet Directory Server and experiencing problems in matching the `issuerDN`, check the Directory Server error logs for useful information.

The second and subsequent lines in the named mapping match properties with values. The `certmap.conf` file has six default properties (you can use the certificate API to customize your own properties):

- `DNComps` is a list of comma-separated attributes used to determine where in the LDAP directory the server should start searching for entries that match the user's information (that is, the owner of the client certificate). The server gathers values for these attributes from the client certificate and uses the values to form an LDAP DN, which then determines where the server starts its search in the LDAP directory. For example, if you set `DNComps` to use the `o` and `c` attributes of the DN, the server starts the search from the `o=<org>, c=<country>` entry in the LDAP directory, where `<org>` and `<country>` are replaced with values from the DN in the certificate.

Note the following situations:

- If there isn't a `DNComps` entry in the mapping, the server uses either the `CmapLdapAttr` setting or the entire subject DN in the client certificate (that is, the end-user's information).
- If the `DNComps` entry is present but has no value, the server searches the entire LDAP tree for entries matching the filter.

- `FilterComps` is a list of comma-separated attributes used to create a filter by gathering information from the user's DN in the client certificate. The server uses the values for these attributes to form the search criteria used to match entries in the LDAP directory. If the server finds one or more entries in the LDAP directory that match the user's information gathered from the certificate, the search is successful and the server optionally performs a verification.

  For example, if `FilterComps` is set to use the email and userid attributes (`FilterComps=e,uid`), the server searches the directory for an entry whose values for email and userid match the end user's information gathered from the client certificate. Email addresses and userids are good filters because they are usually unique entries in the directory. The filter needs to be specific enough to match one and only one entry in the LDAP database.

  For a list of the x509v3 certificate attributes, see the following table:

Table 5-2 Attributes for x509v3 Certificates

Attribute	Description
`c`	Country
`o`	Organization
`cn`	Common name
`l`	Location
`st`	State
`ou`	Organizational unit
`uid`	Unix/Linux userid
`email`	Email address

The attribute names for the filters need to be attribute names from the certificate, not from the LDAP directory. For example, some certificates have an `e` attribute for the user's email address; whereas LDAP calls that attribute `mail`.

- `verifycert` tells the server whether it should compare the client's certificate with the certificate found in the LDAP directory. It takes two values: on, and off. You should only use this property if your LDAP directory contains certificates. This feature is useful to ensure your end-users have a valid, unrevoked certificate.
- `CmapLdapAttr` is a name for the attribute in the LDAP directory that contains subject DNs from all certificates belonging to the user. The default for this property is `certSubjectDN`. This attribute isn't a standard LDAP attribute, so to use this property, you have to extend the LDAP schema. For more information, see *Introduction to SSL*.

  If this property exists in the `certmap.conf` file, the server searches the entire LDAP directory for an entry whose attribute (named with this property) matches the subject's full DN (taken from the certificate). If the search doesn't find any entries, the server retries the search using the `DNComps` and `FilterComps` mappings.

  This approach to matching a certificate to an LDAP entry is useful when it's difficult to match entries using `DNComps` and `FilterComps`.
- `Library` is a property whose value is a pathname to a shared library or DLL. You only need to use this property if you create your own properties using the certificate API. For more information, see the *NSAPI Programmer's Guide.*
- `InitFn` is a property whose value is the name of an init function from a custom library. You only need to use this property if you create your own properties using the certificate API.

For more information on these properties, refer to the examples described in "Sample Mappings," on page 119

Creating Custom Properties

You can use the client certificate API to create your own properties. For information on programming and using the client certificate API, see the *NSAPI Programmer's Guide.*.

Once you have a custom mapping, you reference the mapping as follows:

```
<name>:library <path_to_shared_library>
<name>:InitFn <name_of_init_function>
```

For example:

```
certmap default1 o=Netscape Communications, c=US
default1:library /usr/netscape/enterprise/userdb/plugin.so
default1:InitFn plugin_init_fn
default1:DNComps  ou o c
default1:FilterComps l
default1:verifycert on
```

Sample Mappings

The `certmap.conf` file should have at least one entry. The following examples illustrate the different ways you can use the `certmap.conf` file.

Example #1

This example represents a `certmap.conf` file with only one "default" mapping:

```
certmap default default
default:DNComps ou, o, c
default:FilterComps e, uid
default:verifycert on
```

Using this example, the server starts its search at the LDAP branch point containing the entry `ou=<orgunit>, o=<org>, c=<country>` where the text in <> is replaced with the values from the subject's DN in the client certificate.

The server then uses the values for email address and userid from the certificate to search for a match in the LDAP directory. When it finds an entry, the server verifies the certificate by comparing the one the client sent to the one stored in the directory.

Example #2

The following example file has two mappings: one for default and another for the US Postal Service:

```
certmap default default
default:DNComps
default:FilterComps e, uid

certmap usps ou=United States Postal Service, o=usps, c=US
usps:DNComps ou,o,c
usps:FilterComps e
usps:verifycert on
```

When the server gets a certificate from anyone other than the US Postal Service, it uses the default mapping, which starts at the top of the LDAP tree and searches for an entry matching the client's email and userid. If the certificate is from the US Postal Service, the server starts its search at the LDAP branch containing the organizational unit and searches for matching email addresses. Also note that if the certificate is from the USPS, the server verifies the certificate; other certificates are not verified.

CAUTION The issuer DN (that is, the CA's information) in the certificate must be identical to the issuer DN listed in the first line of the mapping. In the previous example, a certificate from an issuer DN that is `o=United States Postal Service,c=US` won't match because there isn't a space between the `o` and the `c` attributes.

Example #3

The following example uses the `CmapLdapAttr` property to search the LDAP database for an attribute called `certSubjectDN` whose value exactly matches the entire subject DN taken from the client certificate.

```
certmap myco ou=My Company Inc, o=myco, c=US
myco:CmapLdapAttr certSubjectDN
myco:DNComps  o, c
myco:FilterComps mail, uid
myco:verifycert on
```

If the client certificate subject is:

```
uid=Walt Whitman, o=LeavesOfGrass Inc, c=US
```

the server first searches for entries that contain the following information:

```
certSubjectDN=uid=Walt Whitman, o=LeavesOfGrass Inc, c=US
```

If one or more matching entries are found, the server proceeds to verify the entries. If no matching entries are found, the server will use `DNComps` and `FilterComps` to search for matching entries. In this example, the server would search for `uid=Walt Whitman` in all entries under `o=LeavesOfGrass Inc, c=US`.

NOTE This example assumes the LDAP directory contains entries with the attribute `certSubjectDN`.

Setting Stronger Ciphers

The Stronger Ciphers option presents a choice of 168, 128, or 56-bit secret key size for access or no restriction. You can specify a file to be served when the restriction is not met. If no file is specified, iPlanet Web Server returns a “Forbidden” status.

If you select a key size for access that is not consistent with the current cipher settings under Security Preferences, iPlanet Web Server displays a popup dialog warning that you need to enable ciphers with larger secret key sizes.

The implementation of the key size restriction is now based on an NSAPI `PathCheck` directive in `obj.conf`, rather than Service `fn=key-toosmall`. This directive is:

```
PathCheck fn="ssl-check" [secret-keysize=<nbits>]
[bong-file=<filename>]
```

where `<nbits>` is the minimum number of bits required in the secret key, and `<filename>` is the name of a file (not a URI) to be served if the restriction is not met.

`PathCheck` returns `REQ_NOACTION` if SSL is not enabled, or if the `secret-keysize` parameter is not specified. If the secret key size for the current session is less than the specified `secret-keysize`, the function returns `REQ_ABORTED` with a status of `PROTOCOL_FORBIDDEN` if `bong-file` is not specified, or else `REQ_PROCEED`, and the “path” variable is set to the `bong-file` `<filename>`. Also, when a key size restriction is not met, the SSL session cache entry for the current session is invalidated, so that a full SSL handshake will occur the next time the same client connects to the server.

NOTE The Stronger Ciphers form removes any Service `fn=key-toosmall` directives that it finds in an object when it adds a `PathCheck fn=ssl-check`.

To Set Stronger Ciphers, perform the following steps:

1. Access the Server Manager and select the server instance from the drop-down list.
2. Click the Virtual Server Class tab.
3. Select a class from the drop-down list and click Manage.

   The Class Manger page appears.
4. Choose the Content Mgmt tab.
5. Select Stronger Ciphers.

6. Choose to edit:
   - from the drop down list
   - by clicking Browse
   - by clicking Wildcard
7. Select the secret key size restriction:
   - 168 bit or larger
   - 128 bit or larger
   - 56 bit or larger
   - No restrictions
8. Enter the file location of the message to reject access.
9. Click OK.
10. Click Apply.
11. Select hard start /restart or dynamically apply

For more information, see *Introduction to SSL*.

Considering Additional Security Issues

There are other security risks besides someone trying to break your encryption. Networks face risks from external and internal hackers, using a variety of tactics to gain access to your server and the information on it.

So in addition to enabling encryption on your server, you should take extra security precautions. For example, put the server machine into a secure room, and don't allow individuals you don't trust to upload programs to your server.

The following sections describe the most important things you can do to make your server more secure:

- Limit Physical Access
- Limit Administration Access
- Choosing Solid Passwords
- Changing Passwords or PINs
- Limiting Other Applications on the Server

- Preventing Clients from Caching SSL Files
- Limiting Ports
- Knowing Your Server's Limits
- Making Additional Changes to Protect Servers

Limit Physical Access

This simple security measure is often forgotten. Keep the server machine in a locked room that only authorized people can enter. This prevents anyone from hacking the server machine itself.

Also, protect your machine's administrative (root) password, if you have one.

Limit Administration Access

If you use remote configuration, be sure to set access control to allow administration from only a few users and computers. If you want your Administration Server to provide end-user access to the LDAP server or local directory information, consider maintaining two Administration Servers and using cluster management, so that the SSL-enabled Administration Server acts as the master server, and the other Administration Server is available for end-users' access.

For more information regarding clusters, see "About Clusters," on page 131.

You should also turn on encryption for the Administration Server. If you don't use an SSL connection for administration, then you should be cautious when performing remote server administration over an unsecure network. Anyone could intercept your administrative password and reconfigure your servers.

Choosing Solid Passwords

You use a number of passwords with your server: the administrative password, the private key password, database passwords, and so on. Your administrative password is the most important password of all, since anyone with that password can configure any and all servers on your computer. Your private key password is next most important. If someone gets your private key and your private key password, they can create a fake server that appears to be yours, or intercept and change communications to and from your server.

A good password is one you'll remember but others won't guess. For example, you could remember *MCi12!mo* as "My Child is 12 months old!" A bad password is your child's name or birthdate.

Creating Hard-to-Crack Passwords

There are some simple guidelines that will help you create a stronger password.

It is not necessary to incorporate all of the following rules in one password, but the more of the rules you use, the better your chances of making your password hard to crack:

- Passwords should be 6-14 characters long. (Mac passwords cannot be longer than 8 characters)
- Do not use the "illegal" characters: *, ", or spaces
- Do not use dictionary words (any language)
- Do not make common letter substitutions, like replacing E with 3, or L with 1
- Include characters from as many of these classes as possible:
    - Uppercase letters
    - Lowercase letters
    - Numbers
    - Symbols

Changing Passwords or PINs

It's a good practice to change your trust database/key pair file password or PIN periodically. If your Administration Server is SSL enabled, this password is required when starting the server. Changing your password periodically adds an extra level of server protection.

You should only change this password on your local machine. For a list of guidelines to consider when changing a password, see "Creating Hard-to-Crack Passwords," on page 124.

Changing Passwords

To change your trust database/key-pair file password for the Administration Server or an server instance, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Select the Change Password link.
3. Select the security token on which you want to change the password from the drop-down list.

   By default this is ‘internal’ for the internal key database. If you have PKCS#11 modules installed, you will see all the tokens listed. Click the Change Password link.

4. Enter your current password.
5. Enter your new password
6. Enter it again.
7. Click OK.
8. For the Server Manager, click Apply, and then Restart for changes to take effect

Make sure your key-pair file is protected. The Administration Server stores key-pair files in the directory `server_root/alias`. Consider making the files and directory readable only to iPlanet servers installed on your computer.

It’s also important to know if the file is stored on backup tapes or is otherwise available for someone to intercept. If so, you must protect your backups as completely as your server.

Limiting Other Applications on the Server

Carefully consider all applications that run on the same machine as the server. It’s possible to circumvent your server’s security by exploiting holes in other programs running on your server. Disable all unnecessary programs and services. For example, the Unix `sendmail` daemon is difficult to configure securely and it can be programmed to run other possibly detrimental programs on the server machine.

Unix and Linux

Carefully choose the processes started from `inittab` and `rc` scripts. Don't run `telnet` or `rlogin` from the server machine. You also shouldn't have `rdist` on the server machine (this can distribute files but it can also be used to update files on the server machine).

Windows NT

Carefully consider which drives and directories you share with other machines. Also, consider which users have accounts or Guest privileges.

Similarly, be careful about what programs you put on your server, or allow other people to install on your server. Other people's programs might have security holes. Worst of all, someone might upload a malicious program designed specifically to subvert your security. Always examine programs carefully before you allow them on your server.

Preventing Clients from Caching SSL Files

You can prevent pre-encrypted files from being cached by a client by adding the following line inside the `<HEAD>` section of a file in HTML:

```
<meta http-equiv="pragma" content="no-cache">
```

Limiting Ports

Disable any ports not used on the machine. Use routers or firewall configurations to prevent incoming connections to anything other than the absolute minimum set of ports. This means that the only way to get a shell on the machine is to physically use the server's machine, which should be in a restricted area already.

Knowing Your Server's Limits

The server offers secure connections between the server and the client. It can't control the security of information once the client has it, nor can it control access to the server machine itself and its directories and files.

Being aware of these limitations helps you understand what situations to avoid. For example, you might acquire credit card numbers over an SSL connection, but are those numbers stored in a secure file on the server machine? What happens to those numbers after the SSL connection is terminated? You should be responsible for securing any information clients send to you through SSL.

Making Additional Changes to Protect Servers

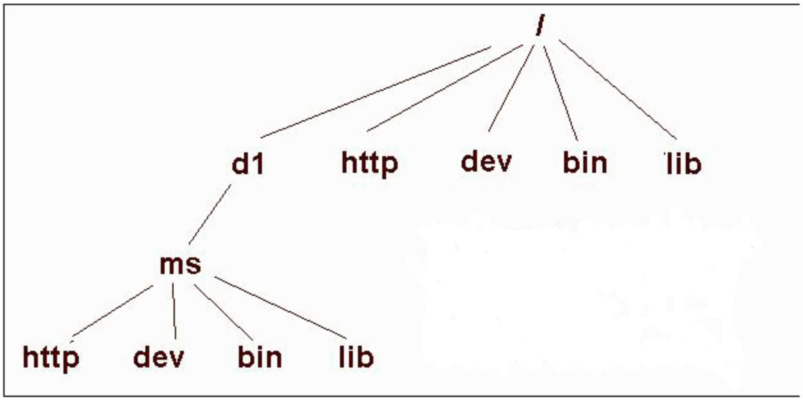
If you want to have both protected and unprotected servers, you should operate the unprotected server on a different machine from the protected one. If your resources are limited and you must run an unprotected server on the same machine as your protected server, do the following.

- Assign proper port numbers. Make sure that the protected server and the unprotected server are assigned different port numbers. The registered default port numbers are:
    - 443 for the protected server
    - 80 for the unprotected server
- For Unix or Linux, enable the `chroot` feature for the document root directory. The unprotected server should have references to its document root redirected using `chroot`.

`chroot` allows you to create a second root directory to limit the server to specific directories. You'd use this feature to safeguard an unprotected server. For example, you could say that the root directory is `/d1/ms`. Then any time the web server tries to access the root directory, it really gets `/d1/ms`. If it tries to access `/dev`, it gets `/d1/ms/dev` and so on. This allows you to run the web server on your Unix/Linux system, without giving it access to all the files under the actual root directory.

However, if you use `chroot`, you need to set up the full directory structure required by iPlanet Web Server under the alternative root directory, as shown in the following illustration:

Figure 5-2 Example of chroot Directory Structure



Specifying chroot for a Virtual Server Class

You can specify the `chroot` directory for a virtual server class by performing the following steps:

1. Access the Server Manager and select the server instance from the drop-down list.
2. Select the Virtual Server Class tab.
3. Click the Edit Classes link.
4. Make sure the Option is set to Edit for the class in which you wish to specify `chroot`.
5. Click the Advanced button for that class.

   The Virtual Servers CGI Settings page appears.
6. Enter the full pathname in the Chroot field.
7. Click OK.
8. Click Apply.
9. Choose Load Configuration Files to dynamically apply.

Specifying chroot for a Virtual Server

You can specify the `chroot` directory for a specific virtual server by performing the following steps:

1. Access the Server Manager and select the server instance from the drop-down list.
2. Select the Virtual Server Class tab.
3. Click on the link for the virtual server you wish to specify the `chroot` directory for from the Tree View of the Server.
4. Select the Settings tab.

   The Settings page appears.
5. Enter the full pathname in the Set to field next to Chroot Directory.
6. Click OK.
7. Click Apply.
8. Choose Load Configuration Files to dynamically apply.

You can also specify the `chroot` directory for a virtual server using the Class Manager Virtual Servers tab and the CGI Settings link.

For more information regarding how to specify a `chroot` directory for a virtual server, see the *Programmer's Guide for iPlanet Web Server*.

Chapter 6

Managing Server Clusters

This chapter describes the concept of clustering iPlanet Web server s and explains how you can use them to share configurations among servers.

This chapter includes the following sections:

- About Clusters
- Guidelines for Using Server Clusters
- Setting Up a Cluster
- Adding a Server to a Cluster
- Modifying Server Information
- Removing Servers from a Cluster
- Controlling Server Clusters
- Adding Variables

About Clusters

A cluster is a group of iPlanet Web Servers that can be administered from a single Administration Server. Each cluster must include one server designated as the administration server. If you have more than one cluster, you can administer all clusters from a single “master” Administration Server. The master administration server retrieves the information about all the clusters and provides the interface for managing the iPlanet Web Servers installed in their respective clusters.

Here are some of the tasks you can accomplish by organizing your servers into clusters:

- Create a central place for administering all iPlanet Web Servers
- Share one or more configuration files between servers
- Start and stop all servers from one “master” Administration Server
- View the access and error logs for the servers you selected

By clustering your iPlanet Web Servers, you’re able to specify a master Administration Server for administering all of your clusters.

NOTE The individual servers can be installed on any computer in a network, but the Administration Server that you designate as the “master” contains information about all clustered servers, and must have access to each cluster’s individual Administration Server.

Guidelines for Using Server Clusters

When you configure a cluster, the master Administration Server containing the information about all clusters communicates with each individual cluster’s Administration Server. The administration server for each cluster must be given the same administration user name and password that the master Administration Server will have.

Before you can create a cluster, all of the servers you want to include in the cluster must be installed. For example, if you want three clusters of five iPlanet Web Servers per cluster, you would need to:

1. Install all of the servers on the computers where they’ll run using the same administration user name and password as the master Administration Server.
2. Configure one of the iPlanet Web Servers in each cluster as the Administration Server.
3. Configure one single cluster’s administration server as the master Administration Server for all clusters. It doesn’t matter which server you choose as the master administration server.

CAUTION Clusters can only be homogeneous. All servers in the cluster must be either Unix or NT. Combining Unix and NT servers in the same cluster may cause the server to hang or crash.

The following list provides some guidelines for configuring groups of servers into clusters:

- Install all of the servers you want to include in a particular cluster prior to creating any clusters.
- Make sure all servers in a cluster must be version 6.0 iPlanet Web Servers.
- Make sure all cluster-specific Administration Servers have the same userid and password as the master administration server. You can use distributed administration to set up multiple administrators on each Administration Server.
- Install servers on any computer in a network, as long as all computers in the cluster are NT or Unix.
- You can designate any cluster-specific Administration Server as the master administration server.
- Make sure the master Administration Server has access to each cluster-specific Administration Server. The master Administration Server retrieves information about all installed iPlanet Web Servers.
- Make sure all Administration Servers are iPlanet Web Server version 6.0 and use the same protocol, HTTP or HTTPS. Only iPlanet Web Server 6.0 servers are supported for addition to clusters.
- If you change the protocol of one Administration Server in a cluster, you must change the protocols for all Administration Servers. Then use the Modify Server interface to modify the individual servers in the cluster.

Setting Up a Cluster

To set up a iPlanet Web Server cluster, perform the following steps:

1. Install the iPlanet Web Servers on the computers you want to include in the cluster.

   Make sure the Administration Server for the cluster has a username and password that the master Administration Server can use for authentication. You can do this either by using the default username and password or by setting up distributed administration.

2. Install the server that will contain the master Administration Server, making sure the username and password matches the one set in Step 1.

3. Add a server to the cluster list.

4. Administer a remote server by accessing its Server Manager forms from the cluster form or by copying a configuration file from one server in the cluster to another.

NOTE After changing the configuration for a remote server, restart the remote server.

Adding a Server to a Cluster

When you add a server to a cluster, you specify its Administration Server and port number. If that Administration Server contains information about more than one server, all of its servers are added to the cluster. You can remove individual servers later.

NOTE If a remote Administration Server contains information about a cluster, the servers in the remote cluster are not added. The master Administration Server adds only those servers that are physically installed on the remote computer.

To add a remote server to a cluster, perform the following steps:

1. Make sure the master Administration Server is tuned on.
2. Access the Administration Server and choose the Cluster Mgmt tab.
3. Click the Add Server link.
4. Choose the protocol that the remote Administration Server uses.
    - `http` for a normal Administration Server
    - `https` for a secure Administration Server
5. Enter the fully qualified domain name as it appears in the `magnus.conf` file of the remote server in the Admin Server Hostname field.

    For example: jodib.iplanet.com
6. Enter the port number for the remote Administration Server.

7. Click OK.

   Your master Administration Server now attempts to contact the remote server. This can take a few minutes. You will receive a message confirming the server is added to the cluster.

8. Click OK.

NOTE If you have two or more servers on different computers that use the same identifier, the server identifier and the hostname for each computer are displayed. When both server identifier and hostnames are the same, the port number is also displayed.

Modifying Server Information

Use the Modify Server option only to update slave administration port information, after it has been changed on the slave server. If you change the port number of a remote Administration Server in your cluster, you also need to modify the information about that Administration Server stored in the cluster. Any other changes to the slave administration server require you to delete the server, and then add it back into the cluster after the changes have been made.

The remote administration servers will not be affected by modification to the master cluster database, unless their files have been transferred through Cluster Control.

To modify information about a server in a cluster, perform the following steps:

1. Go to the master Administration Server and choose the Cluster Mgmt tab.
2. Click the Modify Server link.

   All servers appear listed by their unique server identifier.

3. Select the server or server to modify by:

   ❍ Checking a specific server

   ❍ Clicking Select All

   Click Reset to undo all selections.

4. Enter the new port number.
5. Click OK.

Removing Servers from a Cluster

To remove a server from the cluster, perform the following steps:

1. Go to the master Administration Server and choose the Cluster Mgmt tab.
2. Click the Remove Server link.
3. Select the remote server or servers to modify by:
   - Checking a specific server
   - Clicking Select All

   Click Reset Selection to undo all selections.
4. Click OK.

A message appears confirming that the server is removed from the cluster. You can no longer access the removed server through the cluster; you can only access it now through it's own Administration Server.

Controlling Server Clusters

iPlanet Web Server 6.0 allows you to control the remote servers in your cluster by:

- Starting and stopping them
- Viewing their access and error logs
- Transferring configuration files to them.

CAUTION Clusters must be homogeneous. All servers in the cluster must be either Unix or NT. Transferring configuration files from a different platform may cause the server to hang or crash.

To control servers within your cluster, perform the following steps:

1. Go to the Server Manager for the master Administration Server, and choose the Cluster Mgmt tab.
2. Click the Cluster Control link.

3. Select the server or servers to control by:

   - Checking a specific server
   - Clicking Select All to select all of the servers in the cluster

   Click Reset Selection to undo all selections.

4. Select Start or Stop remote servers from the drop down menu.
5. Select View Access or View Error log records from the drop down menu and enter the number of lines you wish to view.
6. To transfer configuration files:

   a. Select the configuration file you want to transfer in the drop down menu

   b. Select server you want to transfer it from in the drop down menu

   c. Click Transfer.

Adding Variables

Variables are used when servers in a cluster need to be configured with different values. These values might be macros to define slaves using different port numbers, or plug-ins to define different `shlib` paths.

Adding variables affects only the master cluster database. The remote administration servers will not be affected unless their files have been transferred through Cluster Control. When variables are defined, the Administration Server can no longer run independently.

To add variables for a remote server within your cluster, perform the following steps:

1. From the master Administration Server, and choose the Cluster Mgmt tab.
2. Click the Add Variables link.
3. Check the specific server you wish to add variables for.
4. In the Name field enter the type of variable you are adding.

   For example: ‘Port’.

5. In the Value field enter the value you are adding.

   For example: if ‘Port’ is entered in the name field, the value would be the port number.

6. Click OK.

   A message appears confirming that the server variable has been added.

7. Click OK.

The variable must also be added to the server's configuration file you are transferring to the slave. For example:

`SERVERPORT $Port` if port was the variable added.

You can set variables with different values for each slave in the configuration file.

Once added, variables can also be edited and deleted using the drop-down Option list in the Add Variables page.

Part 3

Configuring, Monitoring, and Performance Tuning

Chapter 7

Configuring Server Preferences

This chapter describes how to configure server preferences for your iPlanet Web Server.

This chapter contains the following sections:

- Starting and Stopping the Server
- Tuning Your Server for Performance
- Editing the magnus.conf File
- Adding and Editing Listen Sockets
- Choosing MIME Types
- Restricting Access
- Restoring Configuration Settings
- Configuring the File Cache
- Adding and Using Thread Pools

Starting and Stopping the Server

On Unix, some iPlanet Web Server installations may require access to more memory and/or file descriptors than your operating system allows by default. If you are unable to start the server, check the resource limits imposed by your operating system using the `ulimit` command. Your operating system's `ulimit` man page should provide more information.

Once the server is installed, it runs constantly, listening for and accepting HTTP requests.

The status of the server appears in the Server On/Off page. You can start and stop the server using one of the following methods:

- Click the Server On or Server Off in the Server On/Off page.
- Use the Services window in the Control Panel (Windows NT).
- Use `start.` If you want to use this script with `init`, you must include the start command `http:2:respawn:`*`server_root/type-identifier/`*`start -start -i` in `/etc/inittab`. (Unix/Linux)
- Use `stop`, which shuts down the server completely, interrupting service until it is restarted. If you set the `etc/inittab` file to automatically restart (using "`respawn`"), you must remove the line pertaining to the web server in `etc/inittab` before shutting down the server; otherwise, the server automatically restarts. (Unix/Linux)

After you shut down the server, it may take a few seconds for the server to complete its shut-down process and for the status to change to "Off."

If your machine crashes or is taken offline, the server stops and any requests it was servicing may be lost.

NOTE If you have a security module installed with your server, you will be required to enter the appropriate passwords before starting or stopping the server.

NOTE On Unix, some iPlanet Web Server installations may require access to more memory and/or file descriptors than your operating system allows by default. If you are unable to start the server, check the resource limits imposed by your operating system using the `ulimit` command. Your operating system's `ulimit` man page should provide more information.

Setting the Termination Timeout

When the server is off, it stops accepting new connections. Then it waits for all outstanding connections to complete. The time the server waits before timing out is configurable in the `magnus.conf` file, which can be found in *server_root/*`https-`*server_name*`/config/`. By default it is set to 30 seconds. To change the value, add the following line to `magnus.conf`:

```
TerminateTimeout seconds
```

where *seconds* represents the number of seconds the server will wait before timing out.

The advantages to configuring this value is that the server will wait longer for connections to complete. However, because servers often have connections open from nonresponsive clients, increasing the termination timeout may increase the time it takes for the server to shut down.

Restarting the Server (Unix/Linux)

You can restart the server using one of the following methods:

- Automatically restart it from the `inittab` file.

  Note that if you are using a version of Unix/Linux not derived from System V (such as SunOS 4.1.3), you will not be able to use the `inittab` file.

- Automatically restart it with daemons in `/etc/rc2.d` when the machine reboots.
- Restart it manually.

Because the installation scripts cannot edit the `/etc/rc.local` or `/etc/inittab` files, you must edit those files with a text editor. If you do not know how to edit these files, consult your system administrator or system documentation.

Normally, you cannot start an SSL-enabled server with either of these files because the server requires a password before starting. Although you can start an SSL-enabled server automatically if you keep the password in plain text in a file, this is *not* recommended.

CAUTION Leaving the SSL-enabled server's password in plain text in the server's start script is a large security risk. Anyone who can access the file has access to the SSL-enabled server's password. Consider the security risks before keeping the SSL-enabled server's password in plain text.

The server's start script, key pair file, and the key password should be owned by root (or, if a non-root user installed the server, that user account), with only the owner having read and write access to them.

Starting SSL-enabled Servers Automatically

If security risks are not a concern for you, follow these steps to start your SSL-enabled server automatically:

1. Using a text editor, open the start file, which is located in *server_root/https-server_id.*

2. Locate the `-start` line in the script and insert the following:

   ```
   echo "password"|
   ```

   where *password* is the SSL password you have chosen.

   For example, if the SSL password is `netscape`, the edited line might look like this:

   ```
   -start)
          echo "netscape"|./$PRODUCT_BIN -d $PRODUCT_SUBDIR/config
   $@
   ```

Restarting With Inittab (Unix/Linux)

To restart the server using `inittab`, put the following text on one line in the `/etc/inittab` file:

```
http:2:respawn:server_root/type-identifier/start -start -i
```

where *server_root* is the directory where you installed the server, and *type-identifier* is the server's directory.

The `-i` option prevents the server from putting itself in a background process.

You must remove this line before you stop the server.

Restarting With the System RC Scripts (Unix/Linux)

If you use `/etc/rc.local`, or your system's equivalent, place the following line in `/etc/rc.local`:

```
server_root/type-identifier/start
```

Replace *server_root* with the directory where you installed the server.

Restarting the Server Manually (Unix/Linux)

To restart the server from the command line, log in as root if the server runs on ports with numbers lower than 1024; otherwise, log in as root or with the server's user account. At the command-line prompt, type the following line and press Enter:

```
server_root/type-identifier/start
```

where *server_root* is the directory where you installed the server.

You can use the optional parameter `-i` at the end of the line. `The -i` option runs the server in `inittab` mode, so that if the server process is ever killed or crashed, `inittab` will restart the server for you. This option also prevents the server from putting itself in a background process.

NOTE If the server is already running, the `start` command will fail. You must stop the server first, then use the `start` command. Also, if the server startup fails, you should kill the process before trying to restart it.

Stopping the Server Manually (Unix/Linux)

If you used the `etc/inittab` file to restart the server you must remove the line starting the server from `/etc/inittab` and type `kill -1 1` before you try to stop the server. Otherwise, the server restarts automatically after it is stopped.

To stop the server manually, log in as `root` or use the server's user account (if that is how you started the server), and then type the following at the command line:

```
server_root/type-identifier/stop
```

Restarting the Server (Windows NT)

You can restart the server by:

- Using the Services Control Panel to restart any server.
- Using the Services Control Panel to configure the operating system to restart the server or the administration server each time the machine is restarted.

For Windows NT, perform the following steps:

1. In the Control Panel double-click the Services icon.
2. Scroll through the list of services and select the service for your server.
3. Check Automatic to have your computer start the server each time the computer starts or reboots.
4. Click OK.

NOTE You can also use the Services dialog box to change the account the server uses. For more information about changing the account the server uses, see “Changing the User Account (Unix/Linux),” on page 50.

By default, the web server prompts the administrator for the key database password before starting up. If you want to be able to restart an unattended web server, you need to save the password in a `password.conf` file. Only do this if your system is adequately protected so that this file and the key databases are not compromised.

Using the Automatic Restart Utility (Windows NT)

The server is automatically restarted by a server-monitoring utility if the server crashes. On systems that have debugging tools installed, a dialog box with debugging information appears if the server crashes. To help debug server plug-in API programs (for example, NSAPI programs), you can disable the auto-start feature by setting a very high timeout value. You can also turn off the debugging dialog boxes by using the Registry Editor.

Changing the Time Interval (Windows NT)

To change the time interval that elapses between startup and the time the server can automatically restart, perform the following steps:

1. Start the Registry Editor.
2. Select your server's key (in the left side of the Registry Editor window, located in `HKEY_LOCAL_MACHINE\SOFTWARE\Netscape\Enterprise\6.0`).
3. Choose Add Value from the Edit menu. The Add Key dialog box appears.
4. In Value Name, type `MortalityTimeSecs`.
5. Select REG_DWORD from the Data Type pull-down list.
6. Click OK. The DWORD Editor dialog box appears.
7. Type the time interval (in seconds) that will elapse between startup and the time the server can restart automatically.

   The interval can be in binary, decimal, or hexadecimal format.
8. Click the numerical format for the value you entered in the previous step (binary, decimal, or hexadecimal).
9. Click OK.

   The `MortalityTimeSecs` value appears in hexadecimal format at the right side of the Registry Editor window.

Turning Off the Debugging Dialog Box (Windows NT)

If you've installed an application (such as a compiler) that has modified the system debugging settings and the server crashes, you might see a system-generated application error dialog box. The server will not restart until you click OK.

To turn off the debugging dialog box that appears if the server crashes, perform the following steps:

1. Start the Registry Editor.
2. Select the AeDebug key, located in the left side of the Registry window in `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion`.
3. Double-click the Auto value in the right side of the window.

   The String Editor dialog box appears.
4. Change the string value to 1.

Tuning Your Server for Performance

There are two ways to tune the thread limit: through editing the `magnus.conf` file and through the Server Manager.

If you edit the `magnus.conf` file, `RqThrottleMinPerSocket` is the minimum value and `RqThrottle` is the maximum value.

The minimum limit is a goal for how many threads the server attempts to keep in the `WaitingThreads` state. This number is just a goal. The number of actual threads in this state may go slightly above or below this value. The default value is 48. The maximum threads represents a hard limit for the maximum number of active threads that can run simultaneously, which can become a bottleneck for performance. The default value is 512.

If you use the Server Manager, follow these steps:

1. Go to the Preferences tab.
2. Click the Performance Tuning link.
3. Enter the desired value in the Maximum simultaneous requests field.

For additional information, see the online help for the Performance Tuning page.

Editing the magnus.conf File

When the iPlanet Web Server starts up, it looks in a file called `magnus.conf` in the *server_root*/*server_id*/`config` directory to establish a set of global variable settings that affect the server's behavior and configuration. iPlanet Web Server executes all the directives defined in `magnus.conf`. You can edit certain settings in the `magnus.conf` file using the Magus Editor in the Server Manager.

For a complete description of the magnus.conf file and information about editing the file using a text editor, see the *NSAPI Programmer's Guide*.

To access the Magnus Editor, perform the following steps:

1. Access the Server Manager and choose the Preferences tab.
2. Click the Magnus Editor link.
3. Select the settings to edit from the drop-down list and click Manage.

   The Server Manager displays the editor for the settings you specified.
4. Make the desired changes to the settings and click OK.

For more information about each Settings page, see The Magnus Editor Page in the online help.

Adding and Editing Listen Sockets

Before the server can process a request, it must accept the request via a listen socket, then direct the request to the correct connection group and virtual server. When you install iPlanet Web Server, one listen socket, ls1, is created automatically. This listen socket uses the IP address 0.0.0.0 and the port number you specified as your HTTP server port number during installation (the default is 80). You cannot delete the default listen socket.

You can edit your server's listen socket settings using the Server Manager's Listen Sockets Table. To access the table, perform the following steps:

1. Access the Server Manager and click the Preferences tab.
2. Click the Edit Listen Sockets link.
3. Make the desired changes and click OK.

Choosing MIME Types

The Mime Types page allows you to edit your server's MIME files.

MIME (Multi-purpose Internet Mail Extension) types control what types of multimedia files your mail system supports. MIME types also specify what file extensions belong to certain server file types, for example to designate what files are CGI programs.

You don’t need to create a separate MIME types file for each virtual server. Instead, you create as many MIME types files as you need and associate them with a virtual server. One MIME types file, `mime.types`, exists by default on the server, and cannot be deleted. This file can be the absolute path.

To access the MIME Types page, perform the following steps:

1. Access the Server Manager and click the Preferences tab.

2. Click the MIME Types link.

3. Make the desired changes and click OK.

For more information, see The Mime Settings Page in the online help and Chapter 13, “Using Virtual Servers.”

Restricting Access

You can control access to the entire server or to parts of the server (that is, directories, files, file types) using the Server Manager’s Restrict Access page. When the server evaluates an incoming request, it determines access based on a hierarchy of rules called access-control entries (ACEs), and then it uses the matching entries to determine if the request is allowed or denied. Each ACE specifies whether or not the server should continue to the next ACE in the hierarchy. The collection of ACEs is called an access-control list (ACL). When a request comes in to the server, the server looks in *vsclass*`.obj.conf` (where *vsclass* is the virtual server class name) for a reference to an ACL, which is then used to determine access. By default, the server has one ACL file that contains multiple ACLs.

You can set access control globally for all servers through the Administration Server or for a resource within a specific server instance through the Server Manager. For more information about setting access control for a resource, see “Setting Access Control,” on page 164 in Chapter 8, “Controlling Access to Your Server.”

NOTE You must turn on distributed administration before you can restrict server access.

To restrict access to your iPlanet Web Servers, perform the following steps:

1. Access the Server Manager and choose the Preferences tab.

2. Click the Restrict Access link.

For more information, see Chapter 8, “Controlling Access to Your Server” and The Restrict Access Page in the online help.

Restoring Configuration Settings

The Restore Configuration page allows you to view a backup copy of your configuration files and revert to the configuration data saved on a specific date.

NOTE	On Windows NT, use this page only to roll back your own changes to the configuration files. Do not roll back to backup versions created during installation; they may not be complete.

For more information, see The Restore Configuration Page in the online help.

Configuring the File Cache

The iPlanet Web Server uses a file cache to serve static information faster. In the previous version of the server, there was also an accelerator cache which routed requests to the file cache, but the accelerator cache is no longer used. The file cache contains information about files, and static file content. The file cache also caches information that is used to speed up processing of server-parsed HTML.

The file cache is turned on by default. The file cache settings are contained in a file called `nsfc.conf`. You can use the Server Manager to change the file cache settings.

For more information, see the online *Performance Tuning and Sizing Guide* on `http://docs.iplanet.com/docs/manuals/enterprise.html`.

Adding and Using Thread Pools

You can use thread pools to allocate a certain number of threads to a specific service.

Another use for thread pools is for running thread-unsafe plugins. By defining a pool with the maximum number of threads set to 1, only one request is allowed into the specified service function.

When you add a thread pool, the information you specify includes the minimum and maximum number of threads, the stack size, and the queue size.

For more information, see the online *Performance Tuning and Sizing Guide* on `http://docs.iplanet.com/docs/manuals/enterprise.html`.

The Native Thread Pool and Generic Thread Pools (Windows NT)

On Windows NT, you can use two types of thread pools: the native thread pool (`NativePool`) and additional generic thread pools.

To edit the native thread pool, access the Native Thread Pool page in the Server Manager.

You can create as many generic thread pools as you want, for as many purposes as you want. To create generic thread pools, access the Generic Thread Pools page in the Server Manager.

Thread Pools (Unix/Linux)

Since threads on Unix/Linux are always OS-scheduled (as opposed to user-scheduled) Unix/Linux users do not need to use the `NativePool`, and do not have a Server Manager page for editing its settings. However, Unix/Linux users can still create thread pools. To create thread pools, access the Thread Pools page in the Server Manager.

Editing Thread Pools

Once you have added a thread pool, you can change the values of the thread pool settings (minimum threads, maximum threads and so on) through the Server Manager.

You can also edit the thread pool settings in *vsclass*`.obj.conf`, where *vsclass* is the virtual server class name.

A thread pool appears in *vsclass*`.obj.conf` as follows:

```
Init fn="thread-pool-init" name=name_of_the_pool MaxThreads=n
MinThreads=n QueueSize=n StackSize=n
```

Use the following parameters to change the pool: `MinThreads`, `MaxThreads`, `QueueSize`, and `StackSize`.

Windows NT users can always edit the settings for the native pool using the Server Manager.

Using Thread Pools

After you've set up a thread pool, use it by designating it as the thread pool for a specific service.

To configure a thread pool, go to the Server Manager Preferences tab and select Thread Pool.Once a thread pool is configured, then the Thread Pool list will show the thread pool available to be used for the specific service you've designated

You can also designate a thread pool by using the `pool` parameter of the `load-modules` function in *vsclass*`.obj.conf`, where *vsclass* is the virtual server class name.

```
pool="name_of_pool"
```

In addition, you can use the `pool` parameter on any NSAPI function so that only that NSAPI function runs on the pool you specify.

Chapter 8

Controlling Access to Your Server

This chapter discusses the various methods you can use to control access to the Administration Server and to the files or directories on your web site. For example, for the Administration Server, you can specify who has full control of all the servers installed on a machine and who has partial control of one or more servers. Before you can use access control on the Administration Server, you must enable distributed administration from and set up an administration group in your LDAP database. This chapter assumes you have already configured distributed administration and have defined users and groups in your LDAP database.

You should also ensure the security of the web server as discussed in Chapter 5, “Securing Your Web Server.”

This chapter contains the following sections:

- What Is Access Control?
- How Access Control Works
- Setting Access Control
- Selecting Access Control Options
- Limiting Access to Areas of Your Server
- Working with Dynamic Access Control Files
- Controlling Access for Virtual Servers

What Is Access Control?

Access control allows you to determine:

- Who can access iPlanet Web Administration Server
- Which programs they can access
- Who can access the files or directories on your web site

You can control access to the entire server or to parts of the server, or the files or directories on your web site. You create a hierarchy of rules called access control entries (ACEs) to allow or deny access. Each ACE specifies whether or not the server should check the next ACE in the hierarchy. The collection of ACEs you create is called an access control list (ACL).

By default, the server has one ACL file that contains multiple ACLs. After determining the virtual server to use for an incoming request, iPlanet Web Server checks if any ACLs are configured for that virtual server. If ACLs are found that apply for the current request, iPlanet Web Server evaluates their ACEs to determine whether access should be granted or denied.

You allow or deny access based on:

- Who is making the request (User-Group)
- Where the request is coming from (Host-IP)
- When the request is happening (for example, time of day)
- What type of connection is being used (SSL)

Setting Access Control for User-Group

You can limit access to your web server to certain users or groups. User-Group access control requires users to enter a username and password before gaining access to the server. The server compares the information in a client certificate, or the client certificate itself with a directory server entry.

The Administration Server uses only the basic authenticaton. If you wish to require client authentication on your Administration Server, you must manually edit the ACL files in `obj.conf` changing the method to SSL.

User-Group authentication methods for server instances include:

- Default
- Basic
- SSL
- Digest
- Other

All of these methods require a directory server.

User-Group authentication requires users to authenticate themselves before getting access to the Administration Server, or the files and directories on your web site. With authentication users verify their identity by entering a username and password, using a client certificate, or digest authentication plug-in. Using client certificates requires encryption. For information on encryption and using client certificates, see Chapter 5, “Securing Your Web Server.”

Default Authentication

Default authentication is the preferred method. The Default setting uses the default method you specify in the `obj.conf` file, or “Basic” if there is no setting in `obj.conf`. If you check Default, the ACL rule doesn’t specify a method in the ACL file. Choosing Default allows you to easily change the methods for all ACLs by editing one line in the `obj.conf` file.

Basic Authentication

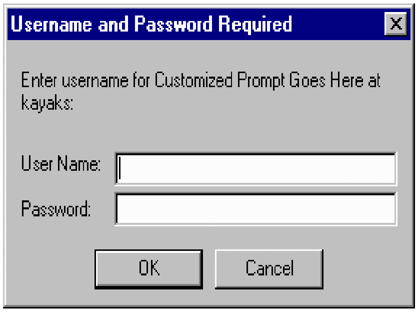
Basic authentication requires users to enter a username and password to access your web server or web site. It is the default setting. You must create and store a list of users and groups in an LDAP database, such as the iPlanet Directory Server. You must use a direstory server installed on a different server root than your web server, or a directory server installed on a remote machine.

When users attempt to access a resource that has User-Group authentication in the Administration Server or on your web site, the web browser displays a dialog box asking the user to enter a username and password. The server receives this information encrypted or unencrypted, depending on whether encryption is turned on for your server.

NOTE Using Basic Authentication without SSL encryption, sends the username and password in unencrypted text across the network. The network packets could be intercepted, and the username and password could be pirated. Basic authentication is most effective when combined with SSL encryption, Host-IP authentication, or both. Using Digest Authentication avoids this problem.

The following dialog appears when users authenticate themselves to the server:

Figure 8-1 Example of Username and Password Prompt



Username and Password Required

Enter username for Customized Prompt Goes Here at kayaks:

User Name:

Password:

After clicking OK, the user will see:

- The Server Administration page, if authenticated to access iPlanet Web Administration Server
- The file or directory listing requested, if logging in to a web site
- A message denying access if the username or password was invalid

You can customize the access denied message that unauthorized users receive in the Access Denied Response page.

SSL Authentication

The server can confirm users' identities with security certificates in two ways:

- Using the information in the client certificate as proof of identity
- Verifying a client certificate published in an LDAP directory (additional)

When you set the server to use certificate information for authenticating the client, the server:

- Checks first if the certificate is from a trusted CA. If not, the authentication fails and the transaction is ended. To learn how to turn on client authentication, see "Requiring Client Authentication," on page 113.
- Maps the certificate to a user's entry using the `certmap.conf` file, if the certificate is from a trusted certificate authority (CA). To learn how to set up the certificate mapping file see "Using the certmap.conf File" on page 116.
- Checks the ACL rules specified for that user if the certificate maps correctly. Even if the certificate maps correctly, ACL rules can deny the user access.

Requiring client authentication for controlling access to specific resources differs from requiring client authentication for all connections to the server. If you set the server to require client authentication for all connections, the client only needs to present a valid certificate issued by a trusted CA. If you set the server's access control to use the SSL method for authentication of users and groups, the client will need to:

- Present a valid certificate issued by a trusted CA
- The certificate must be mapped to a valid user in LDAP
- The access control list must evaluate properly

When you require client authentication with access control, you need to have SSL ciphers enabled for your web server. See Chapter 5, "Securing Your Web Server" to learn how to enable SSL.

In order to successfully gain access to an SSL authenticated resource, the client certificate must be from a CA trusted by the web server. The client certificate needs to be published in a directory server if the web server's `certmap.conf` file is configured to compare the client's certificate in the browser with the client certificate in the directory server. However, the `certmap.conf` file can be configured to only compare selected information from the certificate to the directory server entry. For example, you could configure the `certmap.conf` file to only compare the user ID and email address in the browser certificate with the directory server entry. To learn more about `certmap.conf` and certificate mapping, see Chapter 5, "Securing Your Web Server."

NOTE Only the SSL authentication method requires modification to the `certmap.conf` file, because the certificate is checked against the LDAP directory. Requiring client authentication for all connections to the server does not. If you choose to use client certificates, you should increase the value of the `AcceptTimeout` directive in `magnus.conf`.

Digest Authentication

Digest authentication allows the user to authenticate based on username and password without sending the username and password as cleartext. The browser uses the MD5 algorithm to create a digest value using the user's password and some information provided by the Web Server. This digest value is also computed on the server side using the Digest Authentication plug-in, and compared against the digest value provided by the client. If the digest values match, the user is authenticated.

In order for this to work, your directory server needs access to the user's password in cleartext. iPlanet Directory Server 5.0 includes a reversible password plug-in using a symmetric encryption algorithm to store data in an encrypted form, that can later be decrypted to its original form. Only the Directory Server holds the key to the data.

For digest authentication, you need to enable the reversible password plug-in and the digestauth-specific plug-in included with iPlanet Web Server 6.0. To configure your web server to process digest authentication, set the digestauth property of the database definition in `dbswitch.conf`.

The server tries to authenticate against the LDAP database based upon the ACL method specified, as shown in Table 8-1. If you do not specify an ACL method, the server will use either digest or basic when authentication is required, or basic if authentication is not required. This is the preferred method.

Table 8-1 Digest Authentication Challenge Generation

ACL Method	Digest Authentication Supported by Authentication Database	Digest Authentication Not Supported by Authentication Database
"default"	digest and basic	basic
none specified		
"basic"	basic	basic
"digest"	digest	ERROR

When processing an ACL with `method = digest`, the server attempts to authenticate by:

- Checking for Authorization request header. If not found, a 401 response is generated with a Digest challenge, and the process stops.
- Checking for Authorization type. If Authentication type is Digest the server then:
    - Checks nonce. If not a valid, fresh nonce generated by this server, generates 401 response, and the process stops. If stale, generates 401 response with `stale=true`, and process stops.
    - Checks realm. If it does not match, generates 401 response, and process stops.
    - Checks existence of user in LDAP directory. If not found, generates 401 response, and process stops.
    - Gets request-digest value from directory server and checks for match to client's request-digest. If not, generates 401 response, and process stops.
    - Constructs Authorization-Info header and inserts into server headers.

Installing the Digest Authentication Plug-in on Unix

The Digest Authentication plug-in consists of a shared library found in both:

- `libdigest-plugin.lib`
- `libdigest-plugin.ldif`

To install the Digest Authentication plug-in on Unix, perform the following steps:

1. Make sure this shared library resides on the same server machine that the iPlanet Directory Server is installed on.
2. Make sure you know the Directory Manager password.
3. Modify the `libdigest-plugin.ldif` file changing all references to `/path/to` to the location where you installed the digest plug-in shared library.
4. To install the plug-in, enter the command:

```
% ldapmodify -D "cn=Directory Manager" -w password -a <
libdigest-plugin.ldif
```

Installing the Digest Authentication Plug-in on NT

You will need to copy several `.dll` files from the iPlanet Web Server installation to your iPlanet Directory Server server machine in order for iPlanet Directory Server to start properly with the Digest plug-in.

To install the Digest Authentication plug-in on NT, perform the following steps:

1. Access the shared libraries in the iPlanet Web Server installation in:

   [*server_root*]`\bin\https\bin`

2. Copy the files:
   - `nsldap32v50.dll`
   - `libspnr4.dll`
   - `libplds4.dll`
3. Paste them into either:
   - `\Winnt\system32`
   - iPlanet Directory Server install directory: [*server_root*]`\bin\sldap\server`

Setting the iPlanet Directory Server to Use the DES Algorithm

The DES algorithm is needed to encrypt the attribute where the digest password is stored.

To set the iPlanet Directory Server to use the DES algorithm, perform the following steps:

1. Launch the iPlanet Directory Server Console.
2. Open your iDS 5.0 instance.
3. Select the Configuration tab.
4. Click on the + sign next to plug-ins.
5. Select the DES plug-in.
6. Choose Add to add a new attribute.
7. Enter `iplanetReversiblePassword`.
8. Click Save.
9. Restart your iPlanet Directory Server instance.

NOTE In order to set a digest authentication password in the `iplanetReversiblePassword` attribute for a user, your entry must include the `iplanetReversiblePasswordobject` object.

Other Authentication

You can create a custom authentication method using the access control API.

Setting Access Control for Host-IP

You can limit access to the Administration Server, or the files and directories on your web site by making them available only to clients using specific computers. You specify hostnames or IP addresses for the computers that you want to allow or deny. You can use wildcard patterns to specify multiple computers or entire networks. Access to a file or directory using Host-IP authentication appears seamless to the user. Users can access the files and directories immediately without entering a username or password.

Since more than one person may use a particular computer, Host-IP authentication is more effective when combined with User-Group authentication. If both methods of authentication are used, a username and password will be required for access.

Host-IP authentication does not require DNS to be configured on your server. If you choose to use Host-IP authentication, you must have DNS running in your network and your server must be configured to use it. You can enable DNS on your server through the Performance Tuning page in the Preferences tab on your Server Manager.

Enabling DNS degrades the performance of iPlanet Web Server since the server is forced to do DNS look-ups. To reduce the effects of DNS look-ups on your server's performance, resolve IP addresses only for access control and CGI instead of resolving the IP address for every request. To do this, `iponly=1` to `AddLog fn="flex-log" name="access"` in your `obj.conf` file:

```
    AddLog fn="flex-log" name="access" iponly=1
```

Using Access Control Files

When you use access control on the Administration Server or the files or directories on your web site, the settings are stored in a file with the extension `.acl`. Access control files are stored in the directory *server_install*`/httpacl` with *server_install* being the location where the server is installed. For example, if you installed the server in `/usr/iPlanet/Servers`, the ACL files for both the Administration Server and each server instance configured on your server would be located in `/usr/iPlanet/Servers/httpacl/`.

The main ACL file name is `generated-https-`*server-id*`.acl`; the temporary working file is called `genwork-https-`*server-id*`.acl`. If you use iPlanet Administration Server to configure access, you'll have these two files. However, if you want more complex restrictions, you can create multiple files, and reference them from the `server.xml` file. There are also a few features available only by editing the files such as restricting access to the server based on the time of day or day of the week.

You can also manually create and edit `.acl` files to customize access control using APIs. For more information on using access control APIs, see the *Programmer's Guide*.

For more information on access control files and their syntax, see Appendix C, "ACL File Syntax".

Configuring the ACL User Cache

By default, the iPlanet Web Server caches user and group authentication results in the ACL user cache. You can control the amount of time that ACL user cache is valid by using the `ACLCacheLifetime` directive in the `magnus.conf` file. Each time an entry in the cache is referenced, its age is calculated and checked against `ACLCacheLifetime`. The entry is not used if its age is greater than or equal to the `ACLCacheLifetime`. The default value is 120 seconds. Setting the value to 0 (zero) turns the cache off. If you use a large number for this value, you may need to restart iPlanet Web Server every time you make changes to the LDAP entries. For example, if this value is set to 120 seconds, iPlanet Web Server might be out of sync with the LDAP directory for as long as two minutes. Only set a large value if your LDAP directory is not likely to change often.

Using the `magnus.conf` parameter of `ACLUserCacheSize`, you can configure the maximum number of entries that can be held in the cache. The default value for this parameter is 200. New entries are added to the head of the list, and entries at the end of this list are recycled to make new entries when the cache reaches its maximum size.

You can also set the maximum number of group memberships that can be cached per user entry using the `magnus.conf` parameter, `ACLGroupCacheSize`. The default value for this parameter is 4. Unfortunately non-membership of a user in a group is not cached, and will result in several LDAP directory accesses on every request.

For more information on ACL file directives, see the *NSAPI Programmer's Guide*.

How Access Control Works

When the server gets a request for a page, the server uses the rules in the ACL file to determine if it should grant access or not. The rules can reference the hostname or IP address of the computer sending the request. The rules can also reference users and groups stored in the LDAP directory.

For example, the following ACL file contains the two default entries for the Administration Server (`admin-serv`), plus an additional entry that allows users in the "admin-reduced" group to access the Preferences tab in the Administration Server.

```
version 3.0;
# The following "es-internal" rules protect files such
# as icons and images related to iPlanet Web Server.
# These "es-internal" rules should not be modified.
  acl "es-internal";
  allow (read, list, execute,info) user = "anyone";
  deny (write, delete) user = "anyone";
# The following "default" rules apply to the entire document
# directory of iPlanet Web Server. In this example, the rules
# are set up so that "all" users in the directory server are
# allowed to read, execute, list, and get information.
# The "all" users are not allowed to write to or delete any files.
# All clients that accesses the document directory of the web
# server will be required to submit a username and password
# since this example is using the "basic" method of
# authentication. A client must be in the directory server
# to gain access to this default directory since "anyone"
# not in the directory server is denied, and "all" in the
# directory server are allowed.
  acl "default";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny (all)
  (user = "anyone");
  allow (read,execute,list,info)
  (user = "all");
# The following rules deny access to the directory "web"
# to everyone not in the directory server and deny everyone
# in the directory server who is not in GroupB.
# Only the users in GroupB are allowed read, execute, list,
# and info permissions. GroupA can not gain access to the
# directory "web" even though (in the ACL rule below) they
# can access the directory “my_stuff”. Furthermore, members
# of GroupB can not write or delete files.
  acl "path=/export/user/990628.1/docs/my_stuff/web/";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny (all)
  (user = "anyone");

  allow (read,execute,list,info)
  (group = "GroupB");

# The following rule denies everyone not in the directory
# server and denies everyone in the directory server except
# user with the ID of "SpecificMemberOfGroupB". The ACL rule
# in this setting also has a requirement that the user
# connect from a specific IP address. The IP address setting
# in the rule is optional; it has been added to for extra
# security. Also, this ACl rule has a Customized prompt
# of "Presentation Owner". This Customized prompt appears
# in the username and password dialog box in the client's
# browser.

  acl
"path=/export/user/990628.1/docs/my_stuff/web/presentation.html"
;
  authenticate (user,group) {
     database = "default";
     method = "basic";
     prompt = "Presentation Owner";
  };
  deny (all)
  (user = "anyone" or group = "my_group");
  allow (all)
  (user = "SpecificMemberOfGroupB") and
  (ip = "208.12.54.76");
```

```
# The following ACL rule denies everyone not in the directory
# server and everyone in the directory server except for
# GroupA and GroupB access to the directory “my_stuff”
  acl "path=/export/user/990628.1/docs/my_stuff/";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny (all)
  (user = "anyone");
  allow (read,execute,list,info)
  (group = "GroupA,GroupB");
```

For example, if a user requests the URL:
http://*server_name*/my_stuff/web/presentation.html

iPlanet Web Server would first check access control for the entire server. If the ACL for the entire server was set to continue, the server would check for an ACL for the directory `my_stuff`. If an ACL exists, the server checks the ACEs within the ACL, and then moves on to the next directory. This process continues until an ACL is found that denies access, or until the final ACL for the requested URL (in this case, the file `presentation.html`) is reached.

To set up access control for this example using the Server Manager, you could create an ACL for the file only, or for each resource leading to the file. That is, one for the entire server, one for the `my_stuff` directory, one for the `my_stuff/web` directory, and one for the file.

Setting Access Control

This section describes the process of restricting access to the files or directories on your web site. You can set global access control rules for all servers, and also individually for specific servers. For instance, a human resources department might create ACLs allowing all authenticated users to view their own payroll data, but restrict access to updating data to only human resource personnel responsible for payroll.

You can set access control globally for all servers through the Administration Server. Each option is described in detail in the following section, Selecting the Access Control Options.

NOTE Distributed administration must be configured and activated before global access control can be created.

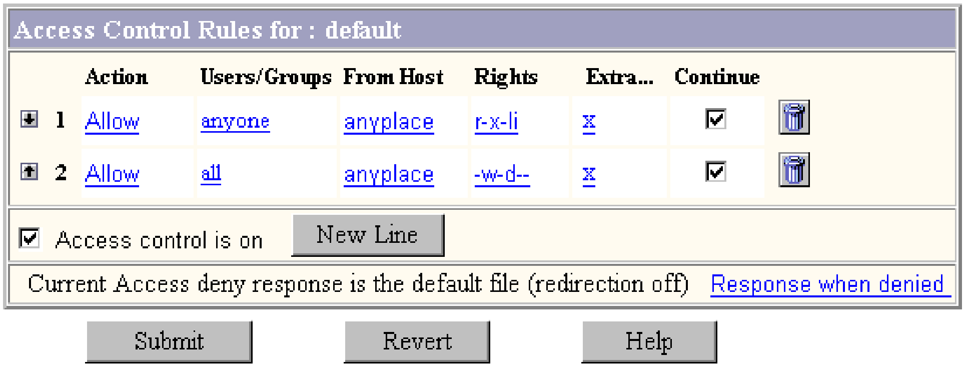
Setting Access Control Globally

To create or edit access control globally for all servers, perform the following steps:

1. Access the Administration Server and choose the Global Settings tab.
2. Click the Restrict Access link.
3. Select the administration server (https-admserv) from the drop-down list.
4. Click either:
   - Create ACL
   - Edit ACL.

   The Access Control Rules for uri=/https-admserv/bin/* page appears:

Figure 8-2 Access Control Rules Page.



Access Control Rules for : default

		Action	Users/Groups	From Host	Rights	Extra...	Continue	
⬇	1	Allow	anyone	anyplace	r-x-li	x	☑	🗑
⬆	2	Allow	all	anyplace	-w-d--	x	☑	🗑

☑ Access control is on

Current Access deny response is the default file (redirection off) Response when denied

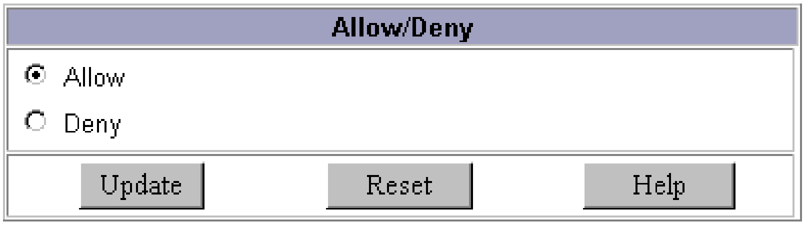
The Administration Server has two lines of default access control rules which cannot be edited.

5. Check Access control is on, if not already selected.

6. To create or edit the global ACL, click on Deny in the Action column.

   The Allow /Deny page is displayed in the lower frame:

Figure 8-3 Allow /Deny Page



Allow/Deny

● Allow

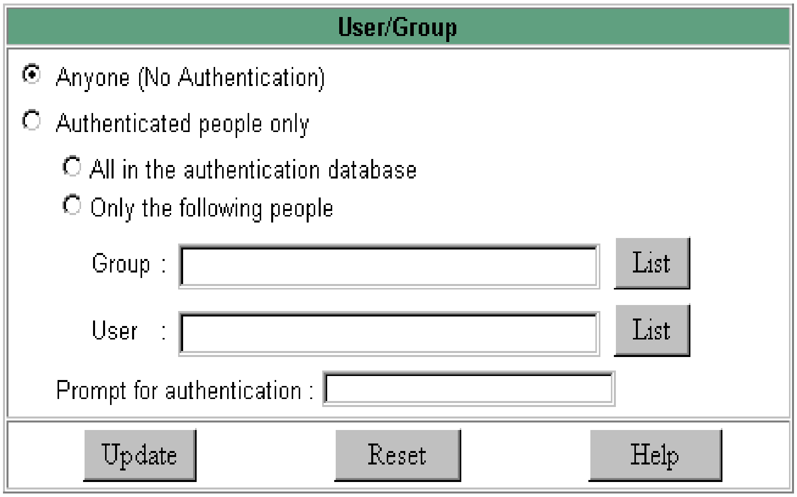
○ Deny

7. Select Allow, if it isn't already selected as the default, and click Update.

8. Click on anyone in the Users/Groups column.

   The User/Group page appears in the lower frame:

Figure 8-4 User/Group Page



User/Group

● Anyone (No Authentication)

○ Authenticated people only

○ All in the authentication database

○ Only the following people

Group :

User :

Prompt for authentication :

9. Select which users and groups you will allow access to and click Update.

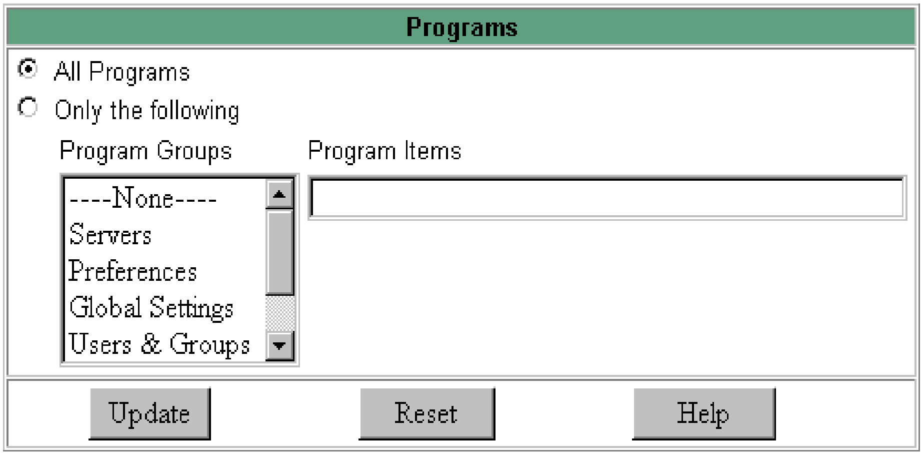
   Clicking List for Group and User will provide lists for you to choose from.

10. Click on anyplace in the From Host column.

11. Enter Host Names and IP Addresses allowed access and click Update.

12. Click on all programs in the Programs column.

Figure 8-5 Programs



Programs

● All Programs

○ Only the following

Program Groups

Program Items

13. Select the Program Groups or enter the specific file name in the Program Items field you will allow access to, and click Update.

14. (Optional) Click the x under the Extra column to add a customized ACL expression.

15. Put a check in the Continue column, if it isn't already selected as the default.

    The server will evaluate the next line before determining if the user is allowed access. When creating multiple lines, work from the most general restrictions to the most specific ones.

16. (Optional) Click Response when denied to direct the user to a different URL or URI.

17. Enter the path to the absolute URL or a relative URI and click update.

18. Click Submit to store the new access control rules in the ACL file.

NOTE Clicking Revert will remove all of the settings you've just created.

Setting Access Control for a Server Instance

You can create, edit, or delete access control for a specific server instance using the Server Manager.

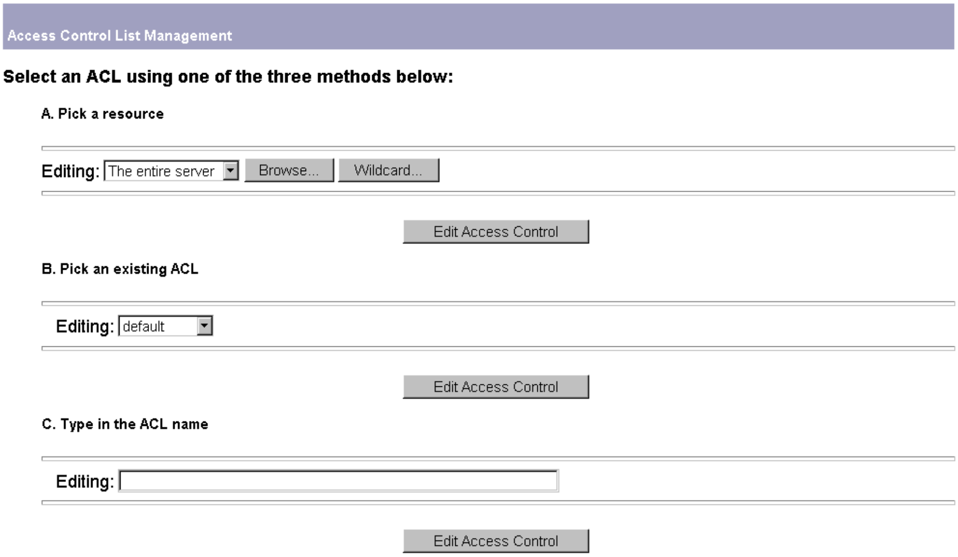
NOTE	If deleting, you should not delete all the ACL rules from the ACL files. At least one ACL file containing a minimum of one ACL rule is required to start the server. Deleting all ACL rules and restarting the server will result in a syntax error.

To create access control for a server instance, perform the following steps:

1. Access the Server Manager and select the server instance you wish to create or edit ACLs for.
2. Choose the Preferences tab from the Server Manager.
3. Click the Restrict Access link.
4. Under the Option column choose one of the following:
   - Add and enter the ACL file location
   - Edit and select the ACL file from the drop-down menu
   - Delete from the drop-down menu and select the ACL file

The Access Control List Management Page offering three options appears:

Figure 8-6 Access Control List Management Page



Access Control List Management

Select an ACL using one of the three methods below:

A. Pick a resource

Editing: The entire server [Browse...] [Wildcard...]

[Edit Access Control]

B. Pick an existing ACL

Editing: default

[Edit Access Control]

C. Type in the ACL name

Editing:

[Edit Access Control]

5. Select one of the following:

   - Pick a resource to specify a wildcard pattern for files or directories (such as `*.html`), choose a directory or a filename to restrict, or browse for a file or directory.
   - Pick an existing ACL to select from a list of all the ACLs you have enabled. Existing ACLs you have not enabled will not appear in this list.
   - Enter the ACL name allows to create named ACLs. Use this option only if you're familiar with ACL files. You'll need to manually edit `obj.conf` if you want to apply named ACLs to resources.

Table 8-2 describes the resource wildcards you can use.

Table 8-2 Server Resource Wildcards

Resource wildcard	What it means
default	A named ACL created during installation that restricts write access so only users in the LDAP directory can publish documents.

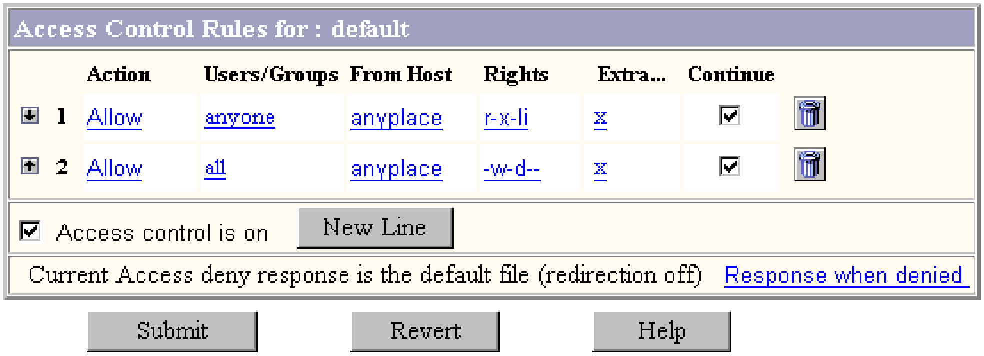
Table 8-2 Server Resource Wildcards

Resource wildcard	What it means
Entire Server	One set of rules determines the access to your entire web site, including any virtual servers you have running. To restrict access to a virtual server, specify the path of its document root.
`/usr/iplanet/server4/docs/cgi-bin/*`	Controls access to all files and directories in the `cgi-bin` directory. You must specify an absolute path. On NT, the path must include the drive letter.
`uri="/sales"`	Controls access to the `sales` directory in the document root. To specify URIs, create a named ACL.

6. Click Edit Access Control.

   The Access Control Rules for: (server instance) appears.

Figure 8-7 Access Control Rules Page



Access Control Rules for : default

		Action	Users/Groups	From Host	Rights	Extra...	Continue	
↓	1	Allow	anyone	anyplace	r-x-li	x	☑	🗑
↑	2	Allow	all	anyplace	-w-d--	x	☑	🗑

☑ Access control is on

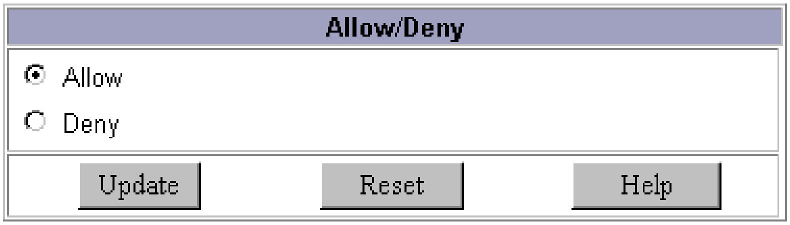
Current Access deny response is the default file (redirection off)

7. Check Access control is on, if not already selected.

8. To create or edit the ACL for this server instance, click on Deny in the Action column.

The Allow /Deny page is displayed in the lower frame:

Figure 8-8 Allow /Deny Page



Allow/Deny

● Allow

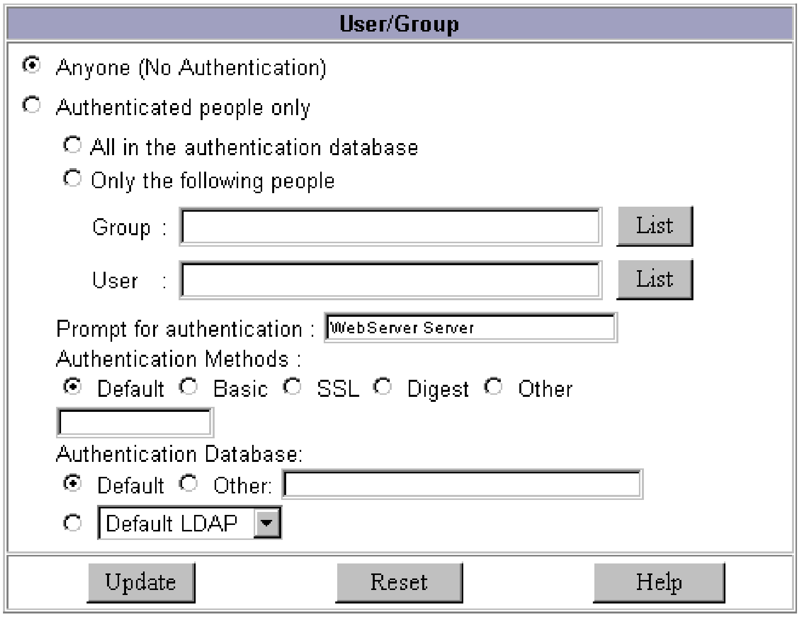
○ Deny

9. Select Allow, if it isn't already selected as the default, and click Update.

10. Click on anyone in the Users/Groups column.

The User/Group page appears in the lower frame:

Figure 8-9 User/Group Page



User/Group

● Anyone (No Authentication)

○ Authenticated people only

○ All in the authentication database

○ Only the following people

Group :

User :

Prompt for authentication : WebServer Server

Authentication Methods :

● Default ○ Basic ○ SSL ○ Digest ○ Other

Authentication Database:

● Default ○ Other:

○ Default LDAP

11. Select which users and groups you will allow access to and click Update.

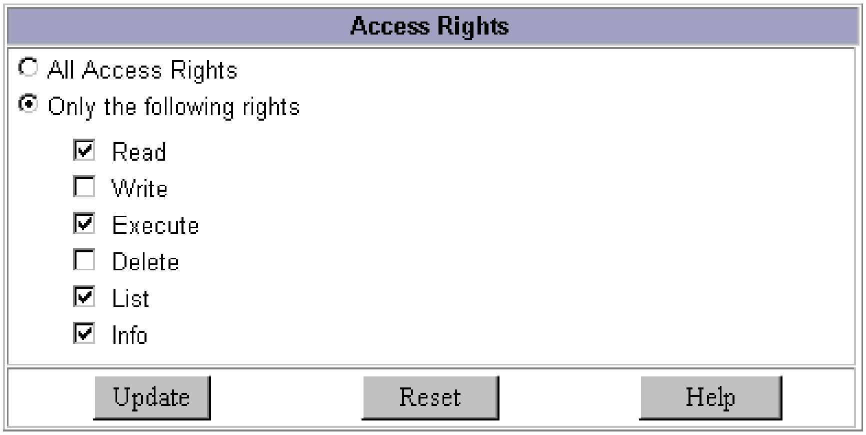
Clicking List for Group and User will provide lists for you to choose from.

12. Click on anyplace in the From Host column.

13. Enter Host Names and IP Addresses allowed access and click Update.

14. Click on all in the Rights column.

Figure 8-10 Access Rights Page



Access Rights

○ All Access Rights

● Only the following rights

☑ Read

☐ Write

☑ Execute

☐ Delete

☑ List

☑ Info

15. Select one of the following and then click Update:

- All Access Rights
- Only the following rights and check all appropriate rights for this user

16. (Optional) Click the x under the Extra column to add a customized ACL expression.

17. Put a check in the Continue column, if it isn’t already selected as the default.

The server will evaluate the next line before determining if the user is allowed access. When creating multiple lines, work from the most general restrictions to the most specific ones.

18. (Optional) Click Response when denied to direct the user to a different URL or URI.

19. Enter the path to the absolute URL or a relative URI and click update.

20. Click Submit to store the new access control rules in the ACL file.

NOTE Clicking Revert will remove all of the settings you’ve just created.

21. Repeat all steps above for each server instance you wish to establish access control for.
22. When finished, click Apply.
23. Select hard start /restart or dynamically apply.

ACL settings can also be enabled on a per virtual server basis. To learn how this is done, see “Editing Access Control Lists for Virtual Servers,” on page 195.

Selecting Access Control Options

The following sections describe the various options that you can select when setting access control. For the Administration Server, the first two lines are set as defaults, and cannot be edited.

Setting the Action

You can specify the action the server takes when a request matches the access control rule.

- **Allow** means users or systems can access the requested resource
- **Deny** means users or systems cannot access the resource

The server goes through the list of access control expressions (ACEs) to determine the access permissions. For example, the first ACE is usually to deny everyone. If the first ACE is set to “continue,” the server checks the second ACE in the list, and if it matches, the next ACE is used. If continue is *not* checked, everyone would be denied access to the resource. The server continues down the list until it reaches either an ACE that doesn’t match, or that matches but is set to not continue. The last matching ACE determines if access is allowed or denied.

Specifying Users and Groups

With user and group authentication, users are prompted to enter a username and password before they can access the resource specified in the access control rule.

iPlanet Web Server checks lists of users and groups stored in an LDAP server, such as iPlanet Directory Server.

You can allow or deny access to everyone in the database, you can allow or deny specific people by using wildcard patterns, or you can select who to allow or deny from lists of users and groups.

- **Anyone (No Authentication)** is the default and means anyone can access the resource without having to enter a username or password. However, the user might be denied access based on other settings, such as host name or IP address. For the Administration Server, this means that anyone in the administrators group that you specified with distributed administration can access the pages.
- **Authenticated people only**
  - **All in the authentication database** matches any user who has an entry in the database.
  - **Only the following people** lets you specify which users and groups to match. You can list users or groups of users individually by separating the entries with commas, or with a wildcard pattern, or you can select from the lists of users and groups stored in the database. Group matches all users in the groups you specify. User matches the individual users you specify. For the Administration Server, the users must also be in the administrators group you specified for distributed administration.
- **Prompt for authentication** allows you to enter message text that appears in the authentication dialog box. You can use this text to describe what the user needs to enter. Depending on the operating system, the user will see about the first 40 characters of the prompt. Netscape Navigator and Netscape Communicator cache the username and password, and associate them with the prompt text. When the user accesses files and directories of the server having the same prompt, the usernames and passwords won't need to be entered again. If you want users to authenticate again for specific files and directories, you simply need to change the prompt for the ACL on that resource.
- **Authentication Methods** specifies the method the server uses for getting authentication information from the client. The Administration server offers only the Basic method of authentication.
  - **Default** uses the default method you specify in the `obj.conf` file, or "Basic" if there is no setting in `obj.conf`. If you check Default, the ACL rule doesn't specify a method in the ACL file. Choosing Default allows you to easily change the methods for all ACLs by editing one line in the `obj.conf` file.
  - **Basic** uses the HTTP method to get authentication information from the client. The username and password are only encrypted if encryption is turned on for the server.

  - ❍ **SSL** uses the client certificate to authenticate the user. To use this method, SSL must be turned on for the server. When encryption is on, you can combine Basic and SSL methods.
  - ❍ **Digest** uses the an authentication mechanism that provides a way for a browser to authenticate based on username and password without sending the username and password as cleartext. The browser uses the MD5 algorithm to create a digest value using the user's password and some information provided by the Web Server. This digest value is also computed on the server side using the Digest Authentication plug-in and compared against the digest value provided by the client.
  - ❍ **Other** uses a custom method you create using the access control API.

- **Authentication Database** lets you select a database the server will use to authenticate users. This option is only available through the Server Manager. If you choose Default, the server looks for users and groups in an LDAP directory. If you wish configure individual ACLs to use different databases, select Other, and choose the database from the drop-down list. Non-default databases and LDAP directories need to have been specified in the file *server_root*/`userdb/dbswitch.conf`. If you use the access control API for a custom database, such as Oracle or Informix, select Other, and enter the database name.

Specifying the From Host

You can restrict access to the Administration Server or your web site based on which computer the request comes from.

- **Anyplace** allows access to all users and systems
- **Only from** allows you to restrict access to specific Host Names or IP Addresses

If you select the Only from option, enter a wildcard pattern or a comma-separated list in the Host Names or IP Addresses fields. Restricting by hostname is more flexible than by IP address: if a user's IP address changes, you won't need to update this list. Restricting by IP address, however, is more reliable: if a DNS lookup fails for a connected client, hostname restriction cannot be used.

You can only use the `*` wildcard notation for wildcard patterns that match the computers' host names or IP addresses. For example, to allow or deny all computers in a specific domain, you would enter a wildcard pattern that matches all hosts from that domain, such as `*.iplanet.com`. You can set different hostnames and IP addresses for superusers accessing the Administration Server.

For hostnames, the * must replace an entire component of the name. That is, *.iplanet.com is acceptable, but *users.iplanet.com is not. When the * appears in a hostname, it must be the left-most character. For example, *.iplanet.com is acceptable, but users.*.com is not.

For the IP address, the * must replace an entire byte in the address. For example, 198.95.251.* is acceptable, but 198.95.251.3* is not. When the * appears in an IP address, it must be the right-most character. For example, 198.* is acceptable, but not 198.*.251.30.

Restricting Access to Programs

Access to programs can only be restricted by the Administration Server. Restricting access to programs allows only specified users to view the Server Manager pages and determines if they can configure that server. For example, you might allow some administrators to configure the Users & Groups section of the administration server and not allow them access to the Global Settings.

- **All Programs** allows or denies access to all programs. By default administrators have access to all programs for a server.
- **Only the following Program Groups** allows you to specify which programs the user has access to. Select the program from the drop-down list. You can choose multiple program groups by pressing the Control key while clicking on the groups. You can restrict access to the following programs groups:
    - None (default)
    - Servers
    - Preferences
    - Global Settings
    - Users & Groups
    - Security
    - Cluster Mgmt

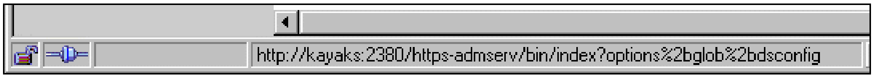
  The Program Groups listed reflect the tabs of the Administration Server, for example, Preferences and Global Settings, and represent access to those pages. When an administrator accesses the Administration Server, the server uses their username, host, and IP to determine what pages they can view.

- **Program Items** allows you to enter a page name in the Program Items field to control access to a specific page within a program.

  To determine the name of a page, place your pointer over the link in the left frame of the Administration Server and then view the text in the status bar on the bottom of your browser. The last word after the last %2b is the name for that page.

  For example, if you want the person who administers an iPlanet Directory Server to have access only to the "Configure Directory Service" page, you would set up a rule that applies only to them (host, IP, and so on), and enter `dsconfig` in the Program Items field

Figure 8-11 Page Name /Program Item



http://kayaks:2380/https-admserv/bin/index?options%2bglob%2bdsconfig

Setting Access Rights

Access rights can only be set by the Server Manager for a server instance. Access rights restrict access to files and directories on your web site. In addition to allowing or denying all access rights, you can specify a rule that allows or denies partial access rights. For example, you allow users read-only access rights to your files, so they can view the information, but not change the files.

- **All Access Rights** is the default and will allow or deny all rights
- **Only the following rights** allows you to select a particular combination of rights to be allowed or denied
  - **Read** allows users to view files, including includes the HTTP methods GET, HEAD, POST, and INDEX
  - **Write** allows users to change or delete files, including the HTTP methods PUT, DELETE, MKDIR, RMDIR, and MOVE. To delete a file, a user must have both write and delete rights
  - **Execute** allows users to execute server-side applications, such as CGI programs, Java applets, and agents
  - **Delete** allows users who also have write privileges to delete files or directories.

- **List** allows users to access lists of the files in directories that don’t contain an `index.html` file.
- **Info** allows users to receive information about the URI, for example `http_head`.

Writing Customized Expressions

You can enter custom expressions for an ACL. Only select this option if you are familiar with the syntax and structure of ACL files. There are a few features available only by editing the ACL file or creating custom expressions. For example, you can restrict access to your server depending on the time of day, day of the week, or both.

The following customized expression shows how you could restrict access by time of day and day of the week. This example assumes you have two groups in your LDAP directory: the “regular” group gets access Monday through Friday, 8:00am to 5:00pm. The “critical” group gets access all the time.

```
allow (read)
{
   (group=regular and dayofweek=”mon,tue,wed,thu,fri”);
   (group=regular and (timeofday>=0800 and timeofday<=1700));
   (group=critical)
}
```

For more information on valid syntax and ACL files, see Appendix C, “ACL File Syntax” and “Referencing ACL Files in obj.conf,” on page 396.

Turning Off Access Control

When you uncheck the option labeled “Access control is on,” you’ll get a prompt asking if you want to erase records in the ACL. When you click OK, the server deletes the ACL entry for that resource from the ACL file.

If you want to deactivate an ACL, you can comment out the ACL lines in the file `generated-https-server-id.acl` by putting # signs at the beginning of each line.

From the Administration Server, you could create and turn on access control for a specific server instance and leave it off (which is the default) for other servers. For example, you could deny all access to the Server Manager pages from the Administration Server. With distributed administration on and access control off by default for any other servers, administrators could still access and configure the other servers, but they cannot configure the Administration Server.

NOTE This access control is in addition to the user being in the administrators group set for distributed administration. The Administration Server first checks that a user (other than superuser) is in the administrators group, and then evaluates the access control rules.

Responding When Access is Denied

iPlanet Web Server provides the following default message when access is denied: “FORBIDDEN. Your client is not allowed access to the restricted object.” You can choose a different response when denied access. You can also create a different message for each access control object.

To change the message sent for a particular ACL, perform the following steps:

1. Click the Response when denied link in the ACL page.
2. Check Respond with the following file in the lower frame.
3. Enter the path to the absolute URL or a relative URI and click update.

   Make sure users have access to the URL or URI they are redirected to.
4. Click Update.
5. Click Submit in the top frame to submit the access control rule.

Limiting Access to Areas of Your Server

This section describes some commonly used access restrictions to a web server and its contents. The steps for each procedure detail the specific actions you need to take; however, you will still need to complete all of the steps described under “Setting Access Control for a Server Instance,” on page 168.

The following procedures are described in this section:

- Restricting Access to the Entire Server
- Restricting Access to a Directory (Path)
- Restricting Access to a URI (Path)
- Restricting Access to a File Type

- Restricting Access Based on Time of Day
- Restricting Access Based on Security

Restricting Access to the Entire Server

You may wish to allow access to users in a group called who access the server from computers in a subdomain. For instance, you may have a server for a company department that you only want users to access from computers in a specific subdomain of your network.

Using the steps described for setting access control for a server instance, you would:

1. Use the Server Manager to select the server instance.
2. Choose the Preferences tab.
3. Click the Restrict Access link.
4. Choose the ACL File to edit.
5. Pick the entire server resource, and click Edit Access Control.
6. Add a new rule to deny access to all.
7. Add another new rule to allow access to a specific group.
8. Enter a wildcard pattern for the host names of the computers to be allowed.

   For example, *.employee.iplanet.com
9. Unselect Continue.
10. Submit and Apply your changes.

Restricting Access to a Directory (Path)

You can allow users in a group to read or run applications in directories, and its subdirectories and files, that are controlled by an owner of the group. For example, a project manager might update status information for a project team to review.

To limit access to a directory on the server, using the steps described for setting access control for a server instance, you would:

1. Use the Server Manager to select the server instance.
2. Choose the Preferences tab.
3. Click the Restrict Access link.

4. Choose the ACL File to edit.
5. Browse the Pick a Resource section and select the directory you want to restrict.

   The directories in the server's document root are displayed. Once selected, the Editing drop-down list displays the absolute path to the directory.

NOTE	If you want to view all files in your server root, click Options and then check List files as well as directories.

6. Click Edit Access Control.
7. Create a new rule and leave the defaults to deny access to everyone from everywhere.
8. Create another new rule allowing users in a specific group to have read and execute rights only.
9. Create a third line to allow a specific user to have all rights.
10. Unselect Continue for the second and third lines and click Update.
11. Submit and Apply your changes.

An absolute path to the file or directory would be created in the docroot directory. The entry in the ACL file would appear as follows:

```
acl “path=d:\iPlanet\suitespot\docroot1\sales/”;
```

Restricting Access to a URI (Path)

You can use a URI to control access to a single user's content on the web server. URIs are paths and files relative to the server's document root directory. Using URIs is an easy way to manage your server's content if you frequently rename or move all or part of it (for example, for disk space). It's also a good way to handle access control if you have additional document roots.

To limit access to a URI, using the steps described for setting access control for a server instance, you would:

1. Use the Server Manager to select the server instance.
2. Choose the Preferences tab.
3. Click the Restrict Access link.

4. Enter the URI you want to restrict in the Type in the ACL name section.

   For example: `uri=/my_directory`.

5. Click Edit Access Control.

6. Create a new rule to allows all users read access.

7. Create another new rule to allow access for the owner of the directory.

8. Uncheck Continue for both the first and second rules.

9. Click Submit and Apply your changes.

A path for the URI is created relative to the document root. The entry in the ACL file would appear as follows: `acl “uri=/my_directory”;`

Restricting Access to a File Type

You can limit access to file types on your server or web site. For example, you might wish to allow only specific users to create programs that run on your server. Anyone would be able to run the programs, but only specified users in the group would be able create or delete them.

To limit access to a file type, using the steps described for setting access control for a server instance, you would:

1. Use the Server Manager to select the server instance.

2. Choose the Preferences tab.

3. Click the Restrict Access link.

4. Click Wildcard in the Pick a resource section and enter a wildcard pattern.

   For example, `*.cgi.`

5. Click Edit Access Control.

6. Create a new rule to allow read access to all users.

7. Create another rule that allows write and delete access only to a specified group.

8. Submit and Apply your changes.

For file type restriction, you would leave both continue boxes checked. If a request for a file comes in, the server will then check the ACL for the file type first.

A Pathcheck function is created in `obj.conf` that may include wildcard patterns for files or directories. The entry in the ACL file would appear as follows: `acl “*.cgi”;`

Restricting Access Based on Time of Day

You can restrict write and delete access to the server or during specified hours or on specified days. You might use this to prevent people from publishing documents during working hours when people might be accessing the files.

To limit access based on time of day, using the steps described for setting access control for a server instance, you would:

1. Use the Server Manager to select the server instance.
2. Choose the Preferences tab.
3. Click the Restrict Access link.
4. Select the entire server from the drop-down list in Pick a Resource and click Edit Access Control.
5. Create a new rule allowing read and execute rights to all.

   This means that if a user wants to add, update, or delete a file or directory, this rule won't apply and the server will search for another rule that matches.
6. Create another new rule denying write and delete rights to all.
7. Click X link to create a customized expression.
8. Enter the days of the week and the times of day to be allowed.

   Example:

   ```
   user = "anyone" and
   dayofweek = "sat,sun" or
   (timeofday >= 1800 and
   timeofday <= 600)
   ```

   The message "Unrecognized expressions" will be displayed in the Users/Groups and From Host fields when you create a custom expression.
9. Submit and Apply your changes.

Any errors in the custom expression will generate an error message. Make corrections and submit again.

Restricting Access Based on Security

As of iPlanet Web Server 6.0 you can configure SSL and non-SSL listen sockets for the same server instance. Restricting access based on security allows you to create protection for resources that should only be transmitted over a secure channel.

To limit access based on security, using the steps described for setting access control for a server instance, you would:

1. Use the Server Manager to select the server instance.
2. Choose the Preferences tab.
3. Click the Restrict Access link.
4. Select the entire server from the drop-down list in Pick a Resource and click Edit Access Control.
5. Create a new rule allowing read and execute rights to all.

   This means that if a user wants to add, update, or delete a file or directory, this rule won’t apply and the server will search for another rule that matches.
6. Create another new rule denying write and delete rights to all.
7. Click X link to create a customized expression.
8. Enter `ssl=”on”`.

   Example:

   ```
   user = "anyone" and ssl=”on”
   ```
9. Submit and Apply your changes.

Any errors in the custom expression will generate an error message. Make corrections and submit again.

Working with Dynamic Access Control Files

Server content is seldom managed entirely by one person. You may need to allow end users to access a subset of configuration options so that they can configure what they need to, without giving them access to the iPlanet Web Server. The subset of configuration options are stored in dynamic configuration files.

The following topics are described in this section:

- Using .htaccess Files
- Supported .htaccess Directives
- .htaccess Security Considerations

Using .htaccess Files

iPlanet Web Server supports`.htaccess` dynamic configuration files. You can enable `.htaccess` files either through the user interface or by manually changing the configuration files. The files that support `.htaccess` are in the *`server_root`*`/plugins/htaccess` directory. These files include a plug-in that enables you to use `.htaccess` files and a script for converting `.nsconfig` files to `.htaccess` files.

You can use `.htaccess` files in combination with the server's standard access control. The standard access controls are always applied before any `.htaccess` access control, regardless of the ordering of `PathCheck` directives. Do not require user authentication with both standard and `.htaccess` access control when user-group authentication is 'Basic' You could use SSL client authentication via the standard server access control, and also require HTTP 'Basic' authentication via an `.htaccess` file.

This section includes the following topics:

- Enabling .htaccess from the User Interface
- Enabling .htaccess from magnus.conf
- Converting Existing .nsconfig Files to .htaccess Files
- Using htaccess-register
- Example of an .htaccess File

Enabling .htaccess from the User Interface

To configure your iPlanet Web Server to use `.htaccess`, perform the following steps:

1. Access the Server Manager and select the server instance you wish to enable `.htaccess` for.
2. Click on the Class Manager link at the top of the screen.
3. Select the Content Mgmt tab.
4. Click on the `.htaccess` Configuration link.

5. Select the server to edit by:
   - Choosing the entire server or a specific server from the drop-down list
   - Choosing the directory and files to edit by clicking Browse
   - Choosing a wildcard pattern to edit by clicking Wildcard
6. Select Yes to activate .htaccess.
7. Enter the file name where you want the `.htaccess` configuration to be added.
8. Click OK.
9. When finished, click Apply.
10. Select hard start /restart or dynamically apply.

Enabling .htaccess from magnus.conf

To manually enable your sever to use the `.htaccess`, you need to first modify the server's `magnus.conf` file to load, initialize, and activate the plug-in.

1. Open `magnus.conf` in the *`server_root`*`/https-`*`server_name`*`/config` file.
2. After the other `Init` directives, add the following lines:
   - For Unix/Linux:

     ```
     Init fn="load-modules" funcs="htaccess-init,htaccess-find"
     shlib="server_root/plugins/htaccess/htaccess.so"
     NativeThread="no"
     Init fn="htaccess-init"
     ```

   - For Windows NT:

     ```
     Init fn="load-modules"
     funcs="htaccess-init,htaccess-find,htaccess-register"
     shlib="server_root/plugins/htaccess/htaccess.dll"
     NativeThread="no"
     Init fn="htaccess-init"
     ```

   - For HP:

     ```
     Initfn="load-modules"
     funcs="htaccess-init,htaccess-find,htaccess-register"
     shlib="<server_root>/plugins/htaccess/htaccess.sl"
     NativeThread="no"
     Init fn="htaccess-init"
     ```

3. (Optional) Make the final line read:

   ```
   Init fn=”htaccess-init”[groups-with-users=yes]
   ```

4. Click File /Save.

5. Open `obj.conf`.

6. Add the PathCheck directive as the last directive in the object.

   a. To activate `.htaccess` file processing for all directories managed by a virtual server, add the `PathCheck` directive to the default object in the `object.conf` file:

      ```
      <Object name="default">
      ...
      ```

   ```
   PathCheck fn="htaccess-find"
   ```

      ```
      </Object>
      ```

   `.htaccess` processing should be the last `PathCheck` directive in the object.

   b. To activate `.htaccess` file processing for particular server directories, place the `PathCheck` directive in the corresponding definition in `magnus.conf`.

7. To name your `.htaccess` files something other than `.htaccess`, you must specify the filename in the `PathCheck` directive using the following format:

   `PathCheck fn="htaccess-find" filename="`*filename*`"`

NOTE The next time you use the Administration Server, you will be warned that manual edits have been applied. Click Apply to accept your changes.

Subsequent access to the server will be subject to `.htaccess` access control in the specified directories. For example, to restrict write access to `.htaccess` files, create a configuration style for them, and apply access control to that configuration style. For more information, see Chapter 17, “Applying Configuration Styles.”

Converting Existing .nsconfig Files to .htaccess Files

iPlanet Web Server 6.0 includes the `htconvert` plug-in for converting your existing `.nsconfig` files to `.htaccess` files. `The` `.nsconfig` files are no longer supported. If you have been using`.nsconfig` files, you should convert them to `.htaccess` files.

When activated, `htconvert` searches the given `server.xml` files for `pfx2dir` and `document-root` directives. Each `.nsconfig` file found will be translated into an `.htaccess` file. Multiple obj.conf files can be converted depending on configuration.

NOTE If there is an existing `.htaccess` file, `htconvert` will produce an .`htaccess.new` file, and give a warning. If `.htaccess` and `.htaccess.new` already exist, the new file will be named .`htaccess.new.new`. The`.new` will be repeatedly appended.

The `htconvert` plug-in currently only supports the `RestrictAccess` and `RequireAuth` directives, and the `<Files>` wrapper. If `<Files>` other than `<Files*>` are presented, the script will give a warning and behave as though all files in the directory are to be access-controlled.

To convert your files, at the command prompt, enter the path to Perl on your system, the path to the plug-in script, and the path to your `server.xml` file. For example:

```
server_root\install\perl server_root/plugins/htaccess/htconvert
server_root/https-server_name/config/server.xml
```

All `.nsconfig` files are converted to `.htaccess` files, but not deleted.

The `groups-with-users` option facilitates handling large numbers of users in groups. If you have many users in a group, follow these steps:

1. Revise the format of the user file format to list all the groups a user belongs to:

   ```
   username:password:group1,group2,group3,...groupn
   ```

2. Revise the `AuthGroupFile` directive to point to the same file as the `AuthUserFile`.

Alternatively, you can:

1. Remove the `AuthGroupFile` directive entirely.
2. Add the following to the `Init fn=htaccess-init` line in the `magnus.conf` file:

   ```
   groups-with-users="yes"
   ```

Using htaccess-register

The `htaccess-register` is a new function allowing you to create your own authentication methods. Like Apache you can create external authentication modules and plug them into the `.htaccess` module via `htaccess-register`. Two sample modules are provided in *`server_root`*`/plugins/nsapi/htaccess`.

You can use external modules to create one or more new directives. For example, you might specify the user database for authentication. The directives may not appear within `<Limit>` or `<LimitExcept>` tags.

Example of an .htaccess File

The following example shows an `.htaccess` file:

```
<Limit> GET POST
order deny,allow
deny from all
allow from all
</Limit>
<Limit> PUT DELETE
order deny,allow
deny from all
</Limit>
AuthName mxyzptlk.kawaii.com
AuthUserFile /server_root/mxyz-docs/service.pwd
AuthGroupFile /server_root/mxyz-docs/service.grp
```

Supported .htaccess Directives

The following `.htaccess` directives are supported in this release:

allow

Syntax

Allows from host where:

- host is all, to allow access from all client hosts
- host is all or the last part of a DNS host name
- host is a full or partial IP address

Does not need to be enclosed within a `<Limit>` or `<LimitExcept>` range but usually is.

Effect

Allows access to the specified hosts. Normally appears inside a `<Limit>` range.

deny

Syntax

Deny from host where:

- host is all, to deny access from all client hosts
- host is all or the last part of a DNS host name
- host is a full or partial IP address

Does not need to be enclosed in a `<Limit> <LimitExcept>` range but usually is.

Effect

Denies access to the specified hosts. Normally appears inside a `<Limit>` range.

AuthGroupFile

Syntax

`AuthGroupFile` filename where filename is the name of file containing group definitions in the form: groupname: user user.

Must not appear within a `<Limit>` or `<LimitExcept>` range.

Effect

Specifies that the named group file is to be used for any group definitions referenced in a require group directive. Note that if the filename specified in an `AuthGroupFile` directive is the same as the filename in an `AuthUserFile` directive, the file is assumed to contain users and groups in the format:

```
username:DES-encrypted-password:comma-separated-list-of-groups
```

AuthUserFile

Syntax

`AuthUserFile` filename where:

- filename is the name of file containing user definitions in the form: `username:password`
- username is a user login name, and password is the DES-encrypted password.

Must not appear within a `<Limit>` or `<LimitExcept>` range.

Effect

Specifies that the named user file is to be used for any user names referenced in a require user or require valid-user directive.

Note that the use of `groups-with-users=yes` in the `Init fn=htaccess-init` directive in `obj.conf`, or specifying an `AuthGroupFile` directive with the same filename, causes that file to be assumed to be in the format:

```
username:DES-encrypted-password:comma-separated-list-of-groups
```

AuthName

Syntax

`AuthName` authentication realm where authentication realm is a string identifying an authorization realm to be associated with any request for user authentication.

Must not appear within a `<Limit>` or `<LimitExcept>` range.

Effect

The authentication realm string typically appears in the prompt for username and password on the client side. It may affect caching of username and password on the client.

AuthType

Syntax

`AuthType` Basic. Must not appear within a `<Limit>` or `<LimitExcept>` range.

Effect

Specifies the user authentication method as HTTP Basic Authentication, the only method currently supported.

<Limit>

Syntax

```
<Limit method method ...>

allow, deny, order, or require directives

</Limit>
```

where method is an HTTP method such as GET, POST, or PUT. Any method that the web server understands can be used here.

Effect

Applies the enclosed directives only for requests using the specified HTTP methods.

<LimitExcept>

Syntax

```
<LimitExcept method method ...>

allow, deny, order, or require directives

</LimitExcept>
```

where method is an HTTP method such as GET, POST, or PUT. Any method that the web server understands can be used here.

Effect

Applies the enclosed directives only for requests types not matching the specified HTTP methods.

order

Syntax

Order ordering where ordering is one of:

- allow, deny
- deny, allow
- mutual-failure

Does not need to be enclosed within a `<Limit>` or `<LimitExcept>` range, but usually is.

Effect

- allows, denies, evaluates allow directives and then deny directives
- denies, allows, evaluates deny directives and then allow directives
- mutual-failure denies access for a host listed in both allow and deny directives, regardless of their ordering

require

Syntax

- requires group groupname groupname
- requires user username username
- requires valid-user

Does not need to be enclosed within a `<Limit>` or `<LimitExcept>` range, but usually is.

Effect

- requires group requires the authenticated user to be a member of one of the specified groups.
- requires user requires the authenticated user to be one of the specified users.
- requires valid-user requires an authenticated user

.htaccess Security Considerations

By default, server support for HTTP PUT is disabled. You can activate HTTP PUT using the Remote File Manipulation page of Content Mgmt in the Class Manager. Great care should be taken in allowing PUT access to directories containing `.htaccess` files, since it will allow them to be replaced. PUT access can be prevented on all files in a directory by restricting access. See “Restricting Access to a Directory (Path),” on page 180.

Controlling Access for Virtual Servers

Access control information in iPlanet Web Server 6.0 can come from a per-virtual server ACL file and `.htaccess` files in the document directories. The `.htaccess` system is unchanged from iPlanet Web Server 4.x.

Your `server.xml` file can contain one or more ACLFILE tags which define an ID associated to a particular standard iPlanet Web Server 6.x ACL file. For example:

```
<ACLFILE id="standard" file="standard.acl">
```

For virtual servers to use access control you must create a reference to one or more ACL file IDs in their ‘aclids’ property. Example:

```
<VS aclids=”standard”>
```

This configuration allows multiple virtual servers to share the same ACL file. If you want to require user-group authentication for a virtual server, you must add one or more USERDB tags to its definition. These USERDB tags create a connection between the database names in your ACL file and the actual databases found in `dbswitch.conf`.

The following example maps the ACLs with no ‘database’ attribute to the ‘default’ database in `dbswitch.conf`:

```
<VS>

   <USERDB id=”default” database=”default”/>

</VS>
```

Accessing Databases from Virtual Servers

You can globally define user authentication databases in the `dbswitch.conf` file. It is only read at server startup.

The `baseDN` of the LDAP URL in `dbswitch.conf` defines the global root of all accesses to the database. This maintains backward compatibility. For most new installations, the `baseDN` would be empty.

`dcsuffix` is a new attribute for LDAP databases in `dbswitch.conf` that defines the root of the DC tree according to the iPlanet LDAP schema. It is relative to the `baseDN` in the LDAP URL. When the `dcsuffix` attribute is present, the LDAP database is iPlanet LDAP schema compliant, and the behaviour of some operations changes. For more information about the iPlanet LDAP schema, and an example see “The iPlanet LDAP Schema” in Chapter 8 of the *NSAPI Programmer’s Guide*.

For every virtual server, you can define one or more `USERDB` blocks that point to one of the directories, and you can define additional information. The `USERDB` blocks ID can be referenced in the database parameter of the ACL. If a virtual server has no `USERDB` blocks, user or group-based ACLs will fail.

`USERDB` tags define an additional layer of indirection between the database attribute of an ACL and `dbswitch.conf`. This layer of indirection adds the necessary protection for the server administrator to have full control over which databases virtual server administrators have access to.

For more information on USERDB, see “User DB Selection” in Chapter 8 of the *NSAPI Programmer’s Guide*.

Specifying LDAP Databases in the User Interface

After you have defined one or more user authentication databases in `dbswitch.conf`, you can use the Class Manager to configure which databases each of your virtual servers will use for authentication. You can also use the Class Manger to add a newly created database definition from `dbswitch.conf` for the virtual server to authenticate against

To specify which LDAP database or databases a virtual server should use, perform the following steps:

1. Access the Server Manager and select the Virtual Server Class tab.
2. Click on the virtual server class link where you wish to specify the LDAP database listed under Tree View of the Server.
3. Select the Virtual Servers tab, if not already displayed.
4. Click the ACL Settings link.
5. Choose ACL Settings from the Select a Setting drop-down list.

   The ACL Settings for Virtual Servers page is displayed.
6. Choose Edit from the drop-down list in the Option column, if not already displayed.
7. Select a database configuration from the drop-down list in the Database column of the virtual server you are editing.
8. Click OK.
9. Close the Edit ACL Files window.
10. Click Apply.
11. Choose dynamically apply.

Editing Access Control Lists for Virtual Servers

ACLs for virtual servers are created for the server instance that the virtual server resides in. Virtual server ACL settings default to those created for the server instance. However, access control for each virtual server can be edited through the Class Manager. You would also use this method to add a newly created ACL file to a virtual server.

To edit ACL settings for a virtual server, perform the following steps:

1. Access the Server Manager and select the Virtual Server Class tab.
2. Click on the virtual server class link where you wish to specify the LDAP database listed under Tree View of the Server.
3. Select the Virtual Servers tab, if not already displayed.
4. Click the ACL Settings link.
5. Choose Edit or Delete from the drop-down list in the Option field for each virtual server you wish to change.
6. Click the Edit link in the ACL File field to display the available ACL files.
7. Select one or more ACL files to add or delete for the virtual server.

   A virtual server can have multiple ACL files because they may have multiple document roots.
8. Choose the database to associate the ACL list with from the drop-down list.
9. (Optional) Enter the BaseDN.
10. Click OK when you have finished making changes.
11. Click Apply.
12. Select dynamically apply.

Chapter 9

Using Log Files

You can monitor your server's activity using several different methods. This chapter discusses how to monitor your server by recording and viewing log files. For information on using the built-performance monitoring services, quality of service features, or SNMP, see Chapter 10, "Monitoring Servers."

This chapter contains the following sections:

- About Log Files
- Viewing an Access Log File
- Viewing the Error Log File
- Archiving Log Files
- Setting Log Preferences
- Running the Log Analyzer
- Viewing Events (Windows NT)

About Log Files

Server log files record your server's activity. You can use these logs to monitor your server and to help you when troubleshooting. The error log file, located in `https-`*server_name*`/logs/errors` in the server root directory, lists all the errors the server has encountered. The access log, located in `https-`*server_name*`/logs/access` in the server root directory, records information about requests to the server and the responses from the server. You can configure the information recorded in the iPlanet Web Server `access` log file. You use the log analyzer to generate server statistics. You can back up server error and access log files by archiving them.

NOTE Due to limitations in the operating system, iPlanet Web Server cannot work with log files larger than 2GB on Linux. As soon as the maximum file size is reached, logging will cease.

Viewing an Access Log File

You can view the server’s active and archived access log files.

To view the Administration Server’s access log from the Administration Server, choose the Preferences tab, and then choose the View Access Log page..

To view an access log for the server instance from the Server Manager, choose the Logs tab, and then choose the View Access Log page

To view an access log for an individual virtual server from the Class Manager, select a virtual server to manage from the highlighted Manage Virtual Servers page, then click the link under the heading Access Log on the Virtual Server Manager page. You can specify the number of entries to view or entries with a conditional qualifier of your choice.

The following is an example of an access log in the Common Logfile Format (you specify the format in the Log Preferences window; see “Setting Log Preferences” on page 202 for more information):

```
wiley.a.com - - [16/Feb/2001:21:18:26 -0800] “GET / HTTP/1.0” 200 751
wiley.a.com - - [17/Feb/2001:1:04:38 -0800] “GET /docs/grafx/icon.gif HTTP/1.0” 204 342
wiley.a.com - - [20/Feb/2001:4:36:53 -0800] “GET /help HTTP/1.0” 401 571
arrow.a.com - john [29/Mar/2001:4:36:53 -0800] “GET /help HTTP/1.0” 401 571
```

Table 9-1 describes the last line of this sample access log.

Table 9-1 The fields in the last line of the sample access log file

Access Log Field	Example
Hostname or IP address of client	arrow.a.com. (In this case, the hostname is shown because the web server’s setting for DNS lookups is enabled; if DNS lookups were disabled, the client’s IP address would appear.

Table 9-1 The fields in the last line of the sample access log file *(Continued)*

Access Log Field	Example
RFC 931 information	- (RFC 931 identity not implemented)
Username	john (username entered by the client for authentication)
Date/time of request	29/Mar/1999:4:36:53 -0800
Request	GET /help
Protocol	HTTP/1.0
Status code	401
Bytes transferred	571

The following is an example of an access log using the flexible logging format (you specify the format in the Log Preferences page; see “Setting Log Preferences” on page 202 for more information):

```
wiley.a.com - - [25/Mar/2001:12:55:26 -0800] "GET /index.htm HTTP/1.0" "GET"
"/?-" "HTTP/   1.0" 304 0 - Mozilla/2.0 (WinNT; I)
wiley.a.com - - [25/Mar/2001:12:55:26 -0800] "GET / HTTP/1.0" "GET" "/?-"
"HTTP/1.0" 304 0    - Mozilla/2.0 (WinNT; I)
wiley.a.com - - [25/Mar/2001:12:55:26 -0800] "GET / HTTP/1.0" "GET" "/?-"
"HTTP/1.0" 304 0    - Mozilla/2.0 (X11; I; IRIX 5.3 IP22)
```

Viewing the Error Log File

The error log file contains errors the server has encountered since the log file was created; it also contains informational messages about the server, such as when the server was started. Unsuccessful user authentication is also recorded in the error log. Use the error log to find broken URL paths or missing files.

To view the Administration Server’s error log file, from the Administration Server, choose the Preferences tab, and choose the View Error Log page.

To view a server instance’s error log file, from the Server Manager, choose the Logs tab, and choose the View Error Log page.

To view an error log for an individual virtual server, from the Class Manager, select a virtual server to manage from the highlighted Manage Virtual Servers page, then click the link under the heading Error Log on the Virtual Server Manager page. You can specify the number of entries to view or entries with a conditional qualifier of your choice.

The following are two examples of entries in the error log; the first example shows an informational message indicating successful start up of the server, the second example indicates that the client `wiley.a.com` requested the file `report.html`, but the file wasn't in the primary document directory on the server.

```
[[22/Jan/2001:14:31:41] info (39700): successful server startup
[22/Jan/2001:14:31:41] info (39700): iPlanet-WebServer-Enterprise/6.0
BB1-01/22/2001 01:45
[22/Jan/2001:14:31:42] warning (13751): for host wiley.a.com trying to GET
/report.html, send-file reports: can't find
/usr1/irenem/ES60-0424/docs/report.html (File not found)
```

Archiving Log Files

You can set up your access and error log files to be automatically archived. At a certain time, or after a specified interval, your logs will be rotated. iPlanet Web Server saves the old log files and stamps the saved file with a name that includes the date and time they were saved. For example, you can set up your access log files to rotate every hour, and iPlanet Web Server saves and names the file "`access.199907152400`," where name of the log file, year, month, day, and 24-hour time is concatenated together into a single character string. The exact format of the log archive file varies depending upon which type of log rotation you set up.

iPlanet Web Server offers the two types of log rotation for archiving files: Internal-daemon log rotation and Cron-based log rotation.

Internal-daemon Log Rotation

This type of log rotation happens within the HTTP daemon, and can only be configured at startup time. Internal daemon log rotation allows the server to rotate logs internally without requiring a server restart. Logs rotated using this method are saved in the following format:

```
access.<4 digit year><2 digit month><2 digit day><4 digit 24-hour time>
error.<4 digit year><2 digit month><2 digit day><4 digit 24-hour time>
```

You can specify the time used as a basis to rotate log files and start a new log file. For example, if the rotation start time is 12:00 a.m., and the rotation interval is 1440 minutes (one day), a new log file will be created immediately upon save regardless of the present time and collect information until the rotation start time. The log file will rotate every day at 12:00 a.m., and the access log will be stamped at 12:00 a.m. and saved as `access.199907152400`. Likewise, if you set the interval at 240 minutes (4 hours), the 4 hour intervals begin at 12:00 a.m. such that the access log files will contain information gathered from 12:00 a.m. to 4:00 a.m., from 4:00 a.m. to 8:00 a.m., and so forth.

If log rotation is enabled, log file rotation starts at server startup. The first log file to be rotated gathers information from the current time until the next rotation time. Using the previous example, if you set your start time at 12:00 a.m. and your rotation interval at 240 minutes, and the current time is 6:00 a.m., the first log file to be rotated will contain the information gathered from 6:00 a.m. to 8:00 a.m, and the next log file will contain information from 8:00 a.m. to 12:00 p.m. (noon), and so forth.

Cron-based Log Rotation

This type of log rotation is based on the time stored in the `cron.conf` file in the *`server_root`*`/https-admserv/config/` directory. This method allows you to archive log files immediately or have the server archive log files at a specific time on specific days. The server's cron configuration options are stored in `ns-cron.conf` in the *`server_root`*`/https-admserv/config/` directory. Logs rotated using the cron based method are saved as the original filename followed by the date and time the file was rotated. For example, `access` might become `access.24Apr-0430PM` when it is rotates at 4:30 p.m.

Log rotation is initialized at server startup. If rotation is turned on, iPlanet Web Server creates a time-stamped access log file and rotation starts at server startup.

Once the rotation starts, iPlanet Web Server creates a new time stamped log file when there is a request or error that needs to be logged to the access or error log file and it occurs after the prior-scheduled "next rotate time".

NOTE You should archive the server logs before running the log analyzer.

To archive log files and to specify whether to use the Internal daemon method or the cron based method, use the Archive Log page in the Server Manager.

Setting Log Preferences

During installation, an access log file named `access` was created for the server. You can customize access logging for any resource by specifying whether to log accesses, what format to use for logging, and whether the server should spend time looking up the domain names of clients when they access a resource.

To use one log file for multiple virtual servers, that log file should have `%vsid%` in its format string for the access log and `LogVsid` should be turned on in the `magnus.conf` file for the error log. Follow the steps below to access the `magnus.conf` file and the logging preferences in the Administration Server UI to make the needed changes. The changes are effected following restart of the Administration server.

To turn on LogVsId:

1. Access the Server Manager and choose the Preferences tab.
2. Click the Magnus Editor link.
3. Select the Logging Settings from the Select a Setting drop-down list and click Manage.
4. Select On for the LogVsId value
5. Click the OK button.

To add %vsid% to the log file format string:

1. Access the Server Manager and choose the Logs tab.
2. Click the Log Preferences link.
3. Enter a new log file location and filename in the Log File: text box.
4. Click the Only Log: radio button.
5. Click the Vsid check box. Alternatively to this, you can click the Custom Format: radio button and add the string ‘%vsid%.

NOTE When adding the custom format string ‘%vsid%, you must use a new access log file.

For information on the LogVsId directive in magnus.conf, see the section "Error Logging and Statistic Collection" in the *NSAPI Programmer's Guide*.

When changing the format of an existing log file, you should first delete/rename the existing log file OR use a different file name.

Server access logs can be in Common Logfile Format, flexible log format, or your own customizable format. The Common Logfile Format is a commonly supported format that provides a fixed amount of information about the server. The flexible log format allows you to choose (from iPlanet Web Server) what to log. A customizable format uses parameter blocks that you specify to control what gets logged. For a list of customizable format parameters, see the *NSAPI Programmer's Guide*.

Once an access log for a resource has been created, you cannot change its format unless you archive it or create a new access log file for the resource.

You can specify logging preferences using the Log Preferences page in the Server Manager, or you can manually configure the following directives in the obj.conf (or magnus.conf) file. In magnus.conf, the server calls the function flex-init to initialize the flexible logging system and the function flex-log to record request-specific data in a flexible log format. To log requests using the common log file format, the server calls init-clf to initialize the Common Log subsystem which is used in obj.conf, and common-log to record request-specific data in the common log format (used by most HTTP servers).

For more information on the NSAPI logging functions, including valid directives and parameters, see the *NSAPI Programmer's Guide*.

Easy Cookie Logging

iPlanet Web Server has an easy way to log a specific cookie using the flexlog facility. Add "Req->headers.cookie.*cookie_name*" to the line that initializes the flex-log subsystem in the configuration file obj.conf. This logs the value of the cookie variable *cookie_name* if the cookie variable is present in the request's headers, and logs "-" if it is not present.

Running the Log Analyzer

The *server-root*/extras/log_anly directory contains the log analysis tool that runs through the Server Manager user interface. This log analyzer analyzes files in common log format only. The HTML document in the log_anly directory that explains the tool's parameters. The *server-root*/extras/flex_anlg directory contains the command-line log analyzer for the flexible log file format. However, the Server Manager defaults to using the flexible log file reporting tool, regardless of whether you've selected common or flexible log file format.

Use the log analyzer to generate statistics about your default server, such as a summary of activity, most commonly accessed URLs, times during the day when the server is accessed most frequently, and so on. You can run the log analyzer from iPlanet Web Server or the command line. The log analyzer cannot generate statistics for virtual servers other than the default server. However, statistics can be viewed for each virtual server as described in "Viewing an Access Log File" on page 198

NOTE Before running the log analyzer, you should archive the server logs. For more information about archiving server logs, see "Archiving Log Files" on page 200.

To run the log analyzer from the Server Manager, follow these steps:

1. From the Server Manager, click the Logs tab.
2. Click Generate Report.
3. Fill in the fields.
4. Click OK.

   The report appears in a new window.

For more information, see the Generate Report Page in the online help.

To analyze access log files from the command line, run the tool, flexanlg, which is in the directory *server-install*/extras/flex_anlg.

To run flexanlg, type the following command and options at the command prompt:

```
flexanlg [ -P ] [-n name] [-x] [-r] [-p order] [-i file]* [ -m
metafile ]* [ o file][ c opts] [-t opts] [-l opts]
```

The following describes the syntax.

```
flexanlg -h.):
-P: proxy log format                                  Default: no
-n servername: The name of the server
-x : Output in HTML                                   Default: no
-r : Resolve IP addresses to hostnames                Default: no
-p [c,t,l]: Output order (counts, time stats, lists)  Default: ctl
-i filename: Input log file(s)                        Default: none
-o filename: Output log file                          Default: stdout
-m filename: Meta file(s)                             Default: none
-c [h,n,r,f,e,u,o,k,c,z]: Count these item(s) -       Default: hnreuokc
    h: total hits
    n: 304 Not Modified status codes (Use Local Copy)
    r: 302 Found status codes (Redirects)
    f: 404 Not Found status codes (Document Not Found)
    e: 500 Server Error status codes (Misconfiguration)
    u: total unique URL’s
    o: total unique hosts
    k: total kilobytes transferred
    c: total kilobytes saved by caches
    z: Do not count any items.
-t [sx,mx,hx, xx,z]: Find general stats - Default:s5m5h24x10
    s(number): Find top (number) seconds of log
    m(number): Find top (number) minutes of log
    h(number): Find top (number) hours of log
    u(number): Find top (number) users of log
    a(number): Find top (number) user agents of log
    r(number): Find top (number) referers of log
    x(number): Find top (number) for miscellaneous keywords
    z: Do not find any general stats.
-l [cx,hx]: Make a list of - Default: c+3h5
    c(x,+x): Most commonly accessed URLs
             (x: Only list x entries)
             (+x: Only list if accessed more than x times)
    h(x,+x): Hosts (or IP addresses) most often accessing your server
             (x: Only list x entries)
             (+x: Only list if accessed more than x times)
    z: Do not make any lists
```

Viewing Events (Windows NT)

In addition to logging errors to the server error log (see "Viewing the Error Log File" on page 199), iPlanet Web Server logs severe system errors to the Event Viewer. The Event Viewer lets you monitor events on your system. Use the Event Viewer to see errors resulting from fundamental configuration problems, which can occur before the error log can be opened.

To use the Event Viewer, perform the following steps:

1. From the Start menu, select Programs and then Administrative Tools. Choose Event Viewer in the Administrative Tools program group.
2. Choose Application from the Log menu.

The Application log appears in the Event Viewer. Errors from iPlanet Web Server has a source label of `https-`*`serverid`* or `WebServer6.0`.

3. Choose Find from the View menu to search for one of these labels in the log. Choose Refresh from the View menu to see updated log entries.

   For more information about the Event Viewer, consult your system documentation.

Chapter 10

Monitoring Servers

This chapter contains information on ways to monitor your server, including the built-in monitoring tool, the quality of service features, and Simple Network Management Protocol (SNMP).

You can use SNMP together with iPlanet management information bases (MIB) and network management software such as HP OpenView to monitor your servers in real-time just as you monitor other devices in your network. If you're using Windows NT, SNMP is built in and already enabled.

You can view the server's status in real time by using the statistics feature or the SNMP. If you're using Unix or Linux, you must configure your iPlanet server for SNMP if you plan to use it. This chapter provides the information you need to use SNMP on Unix or Linux with your iPlanet server.

The following topics are included in this chapter:

- Monitoring the Server Using Statistics
- Using Quality of Service
- SNMP Basics
- The iPlanet Web Server MIB
- Setting Up SNMP
- Using a Proxy SNMP Agent (Unix/Linux)
- Reconfiguring the SNMP Native Agent
- Installing the SNMP Master Agent
- Enabling and Starting the SNMP Master Agent
- Configuring the SNMP Master Agent

- Enabling the Subagent
- Understanding SNMP Messages

Monitoring the Server Using Statistics

You can use the statistics feature to monitor your server's current activity. The statistics show you how many requests your server is handling and how well it is handling these requests. You can view some statistics for individual virtual servers, and others for the entire server instance. If the interactive server monitor reports that the server is handling a large number of requests, you may need to adjust the server configuration or the system's network kernel to accommodate the requests. For more information, see the online *Performance Tuning and Sizing Guide* on `http://docs.iplanet.com/docs/manuals/enterprise.html`.

Once you enable statistics, you can view statistics in the following areas:

- connections
- DNS
- KeepAlive
- cache
- virtual servers

For a description of the various server statistics for which the interactive server monitor reports the totals, see the Monitor Current Activity page in the online help.

CAUTION When you enable statistics/profiling, statistics information will be available to any user of your server. See the description of stats-xml in the *NSAPI Programmer's Guide* for more information.

Enabling Statistics

To enable statistics, follow these steps:

1. From the Server Manager, click the Monitor tab.
2. Click Monitor Current Activity
3. Click Yes to enable statistics.

4. Click OK.
5. Click Apply to apply your changes. You do not need to restart the server.

For more information on enabling statistics, see the online help.

Using Statistics

Once you've enabled statistics, you can get a variety of information on how your server instance and your virtual servers are running. The statistics are broken up into functional areas.

To access statistics, follow these steps:

1. From the Server Manager, click the Monitor tab.
2. Click Monitor Current Activity.
3. From the pull-down list, choose the poll interval.

   The poll interval is the number of seconds between updates of the statistics information displayed.
4. From the pull-down list, choose the kind of statistics you want displayed.
5. Click OK.

   If your server instance is running, and you have enabled statistics/profiling, you see a page displaying the kind of statistics you selected. The page is updated every 5-15 seconds, depending upon what you chose for the poll interval.

You can use the data you see in statistics to tune your server. For more information, see the online *Performance Tuning and Sizing Guide* on `http://docs.iplanet.com/docs/manuals/enterprise.html`.

Using Quality of Service

Quality of Service refers to the performance limits you set for a server instance virtual server class, or virtual server. For example, if you are an ISP, you might want to charge different amounts of money for virtual servers depending on how much bandwidth you allow them. You can limit two areas: the amount of bandwidth and the number of connections.

You can enable these settings for the entire server or for a class of virtual servers in the Server Manager from the Monitor tab. However, you can override these server or class-level settings for an individual virtual server. For more information on setting quality of service limits for an individual server, see “Configuring Virtual Server Quality of Service Settings,” on page 319.

Two settings govern how traffic is counted and how often the bandwidth is recomputed: the recompute interval and the metric interval. The recompute is how often (in milliseconds) the bandwidth is computed. The metric interval is the period of time for which data is used in traffic calculations.

This section includes the following topics:

- Quality of Service Example
- Setting Up Quality of Service
- Required Changes to obj.conf
- Known Limitations to Quality of Service

Quality of Service Example

The following example shows how the quality of service information is collected and computed:

The server has metric interval of 30 seconds.

The server starts up at a time of 0 seconds.

At time 1 second, an HTTP connection generates 5000 bytes of traffic to/from the server.

No more connections are made after that. At 30 seconds, the total traffic for the last 30 seconds is 5000 bytes.

At 32 seconds, the traffic sample from 1 second is discarded, since it is older than the 30 seconds of the metric interval. The total traffic for the last 30 seconds is now 0.

The recompute interval works similarly. The server’s recompute interval is 100ms.

Continuing with the example, the bandwidth gets recomputed periodically every 100 milliseconds. The calculation is based on the amount of traffic as well as the metric interval.

At time 0 seconds, the bandwidth is calculated for the first time. The total traffic is zero, divided by the metric interval of 30 seconds, gives a bandwidth of zero.

At 1 second, the bandwidth is calculated for the 10th time (1000 milliseconds/ 100 milliseconds). The total traffic is 5000 bytes, which is divided by 30 seconds. The bandwidth is 5000/30 = 166 bytes per second.

At 30 seconds, the bandwidth is calculated for the 300th time. The total traffic is 5000 bytes, which is divided by 30 seconds. The bandwidth is 5000/30 = 166 bytes per second.

At 32 seconds, the bandwidth is computed again for the 320th time. The traffic is now 0 (since the one connection that generated traffic is too old to be counted), divided by 30, gives a bandwidth of 0 bytes/second.

Setting Up Quality of Service

To configure the quality of service settings for a server instance or a class of virtual servers, you need to configure the settings in through the user interface. To actually enforce your quality of service settings, you must also set up Server Application Functions (SAFs) in your obj.conf file.

To configure quality of service, follow these steps:

1. From the Server Manager, click the Monitor tab.
2. Click Quality of Service.

   A page appears listing general settings for quality of service, followed by a list containing the server instance as a whole and each class of virtual servers.
3. To enable quality of service as a whole, click Enable.

   By default quality of service is disabled. Enabling quality of service increases server overhead slightly.
4. Choose the Recompute Interval.

   The recompute interval is the number of milliseconds between each computation of the bandwidth for all servers, classes, and virtual servers. The default is 100 milliseconds.
5. Choose the Metric Interval.

   The metric interval is the interval in seconds during which the traffic is measured. The default is 30 seconds. All bandwidth measured during this time is averaged to give the bytes per second.

If your site has a lot of large file transfers, use a large value (several minutes or more) or this field. A large file transfer might take up all the allowed bandwidth for a short metric interval, and result in connections being denied if you've enforced the maximum bandwidth setting. Since the bandwidth is averaged by the metric interval, a longer interval smooths out spikes caused by large files.

If the bandwidth limit is much lower than available bandwidth (for example, 1 MB-per-second bandwidth limit but with a 1 GB-per-second connection to the backbone), the metric interval should be shortened.

Please note that if you have large static file transfers and a bandwidth limit that is much lower than available bandwidth, you have to decide which situation to tune for, since the problems require opposite solutions.

6. Enable quality of service for the server instance and/or the virtual server classes.

   The lower portion of the screen lists the server instance and server classes. Choose Enable as the action next to the items for which you want to enable quality of service.

7. Set the maximum bandwidth, in bytes per second.

8. Choose whether or not to enforce the maximum bandwidth setting.

   If you choose to enforce the maximum bandwidth, once the server reaches its bandwidth limit additional connections are refused.

   If you do not enforce the maximum bandwidth, when the maximum is exceeded the server logs a message to the error log.

9. Choose the maximum number of connections allowed.

   This number is the number of concurrent requests processed.

10. Choose whether or not to enforce the maximum connections setting.

    If you choose to enforce the maximum connections, once the server reaches its limit additional connections are refused.

11. If you do not enforce the maximum connections, when the maximum is exceeded the server logs a message to the error log.

12. Click OK.

Required Changes to obj.conf

To enable quality of service, you must include directives in your `obj.conf` to invoke two Server Application Functions (SAFs): an AuthTrans `qos-handler` and an Error `qos-error`.

The `qos-handler` AuthTrans directive must be the first `AuthTrans` configured in the default object in order to work properly. The role of the quality of service handler is to examine the current statistics for the virtual server, virtual server class, and global server, and enforce the limits by returning an error.

iPlanet Web Server includes a built-in sample quality of service handler SAF, called `qos-handler`. This SAF logs when limits are reached, and returns 503 "Server busy" to the server so that it can be processed by NSAPI.

iPlanet Web Server also includes a built-in sample error SAF called `qos-error` which returns an error page stating which limits caused the 503 error and the value of the statistic that triggered the limit. You may want to alter the sample code to provide different error information.

These samples are available at *server_root*/plugins/nsapi/examples/qos.c. You can use these samples, or you can write your own SAFs.

For more information on these SAFs and how to use them, see the *NSAPI Programmer's Guide*.

Known Limitations to Quality of Service

When you use the quality of service features, keep in mind the following limitations:

- The connection or bandwidth statistics are not shared across server processes because of performance. In other words, the setting of `MaxProc` is not accounted for. So all the limits apply individually to a server process, not to the aggregate of all processes. For more information on `MaxProcs` and multiple processes, see the online *Performance Tuning and Sizing Guide* on `http://docs.iplanet.com/docs/manuals/enterprise.html`.
- The quality of service features only measure the HTTP bandwidth at the application level. The HTTP bandwidth can differ from the actual TCP network bandwidth for a variety of reasons:
    - If SSL is enabled, handshakes and client certificate exchanges add to the traffic but are not measured.
    - If chunked encoding is enabled in either or both directions, the chunking layer removes the chunk headers and they are not counted in the traffic. Other headers or protocol items are counted.
- The quality of service features cannot accurately measure traffic from `PR_TransmitFile` calls. For basic I/O operations such as `PR_Send()/net_write` or `PR_Recv()/net_read`, the data transferred can be quickly accounted for by the bandwidth manager, since the number of bytes transferred in one system call is usually

the size of a buffer and the I/O call returns quickly. This works very well to measure the instantaneous bandwidth of dynamic content applications. However, because the amount of data transferred from `PR_TransmitFile` is only known at the end of the transfer, it can’t be measured before it completes.

If the `PR_TransmitFile` is short, then the quality of service features will perform adequately. However, If the `PR_TransmitFile` is long, such as in the case of a long file downloaded by a dialup user, the whole amount of data transferred will be counted at completion time. When the bandwidth manager recomputes bandwidth after the next recompute interval period starts, the bandwidth computed will go up significantly because of that recent large `PR_TransmitFile`. This case could cause the server to deny all requests until the next metric interval, when the bandwidth manager will "expire" the transmit file operation, since it is too old, and thus the bandwidth value will go back down. If your site has a lot of very long static file downloads , the you should increase the metric interval from the default 30 seconds.

- The bandwidth computed is always an approximation because it is not measured instantaneously, but is recomputed at regular intervals and over a certain period. For example, if the metric interval is the default 30 seconds and the server is idle for 29 seconds, then the next second, a client could potentially use 30 times the bandwidth limit in one second.
- The quality of service bandwidth statistics are lost whenever the server is reconfigured dynamically. In addition, the quality of service limitations are not enforced in threads that have connections on an older, inactive configuration, because the bandwidth manager thread only computes bandwidth statistics for the active configuration. Potentially, a client that doesn’t close its socket for a long time and remains active so that the server doesn’t time it out would not be subject to the quality of service limitations after a server dynamic reconfiguration.
- The concurrent connections are computed with a different granularity for virtual servers than for virtual server classes and the global server instance. The connection counter for an individual virtual server is incremented atomically immediately after the request is parsed and routed to the virtual server. It is also decremented atomically at the end of the response processing for that request. This means that the virtual server connection statistics are always exact at any instant.

  However, the connection statistics for the virtual server class and global server instance are not updated instantly. They are updated by the bandwidth manager thread every recompute interval. The connection count for the virtual server class is the sum of the connections on all virtual servers of that class; and the global server instance connection count is the sum of connections on all virtual server classes.

Because of the way these values are computed, the number of connections for a virtual server is always correct (and if you've enforced a limit to the number of connections, you can never have more than the limit), and the virtual server class and server instance values are not quite as accurate, since they're only computed at intervals.

SNMP Basics

SNMP is a protocol used to exchange data about network activity. With SNMP, data travels between a managed device and a network management station (NMS). A managed device is anything that runs SNMP: hosts, routers, your web server, and other servers on your network. The NMS is a machine used to remotely manage that network. Usually, the NMS software will provide a graph to display collected data or use that data to make sure the server is operating within a particular tolerance.

The NMS is usually a powerful workstation with one or more network management applications installed. A network management application such as HP OpenView graphically shows information about managed devices, such as your web servers. For example, it might show which servers in your enterprise are up or down, or the number and type of error messages received. When you use SNMP with an iPlanet server, this information is transferred between the NMS and the server through the use of two types of agents, the subagent and the master agent.

The subagent gathers information about the server and passes the information to the server's master agent. Every iPlanet server, except for the Administration Server, has as subagent.

NOTE After making any SNMP configuration changes, you must click the Apply button, then restart SNMP subagent.

The master agent exchanges information between the various subagents and the NMS. The master agent is installed with the Administration Server.

You can have multiple subagents installed on a host computer, but only one master agent. For example, if you had Directory Server, iPlanet Web Server, and the Messaging Server installed on the same host, the subagents for each of the servers would communicate with the same master agent.

The iPlanet Web Server MIB

iPlanet Web Server stores variables pertaining to network management. Variables the master agent can access are called managed objects. These objects are defined in a tree-like structure called the management information base (MIB). The MIB provides access to the web server's network configuration, status, and statistics. Using SNMP, you can view this information from the network management workstation (NMS).

The top level of the MIB tree shows that the internet object identifier has four subtrees: directory (1), mgmt (2), experimental (3), and private (4). The private (4) subtree contains the enterprises (1) node. Each subtree in the enterprises (1) node is assigned to an individual enterprise, which is an organization that has registered its own specific MIB extensions. An enterprise can then create product-specific subtrees under its subtree. MIBs created by companies are located under the enterprises (1) node. The iPlanet MIBs are located under the enterprises (1) node.

Each iPlanet server subagent provides a MIB for use in SNMP communication. The server reports significant events to the network management station (NMS) by sending messages or traps containing these variables. The NMS can also query the server's MIB for data, or can remotely change variables in the MIB.

Each iPlanet server has its own management information base (MIB). All iPlanet MIBs are located at:

server_root/`plugins/snmp`

The iPlanet Web Server's MIB is a file called `iws.mib`. This MIB contains the definitions for various variables pertaining to network management for iPlanet Web Server.

The iPlanet Web Server 6.0 MIB has an object identifier of `http 60 (iws60 OBJECT IDENTIFIER ::= {http 60 })` and is located in the *server_root*/`plugins/snmp` directory.

You can see administrative information about your web server and monitor the server in real time using the iPlanet Web Server MIB. Table 10-1 lists and describes the managed objects stored in the `iws.mib`.

Table 10-1 `iws.mib` managed objects and descriptions

Managed object	Description
`iwsInstanceTable`	iPlanet Web Server instances.
`iwsInstanceEntry`	iPlanet Web Server instance.
`iwsInstanceIndex`	Server instance index.
`iwsInstanceId`	Server instance identifier

Table 10-1 `iws.mib` managed objects and descriptions *(Continued)*

Managed object	Description
`iwsInstanceVersion`	String, such as `iPlanet-WebServer-Enterprise/ 6.0 BB1-01/24/2001 17:15 (SunOS DOMESTIC)`
`iwsInstanceDescription`	Description of the server instance.
`iwsInstanceOrganization`	Organization responsible for the server instance.
`iwsInstanceContact`	Contact information for person(s) responsible for server instance.
`iwsInstanceLocation`	Where the server is located.
`iwsInstanceStatus`	Status of the server instance.
`iwsInstanceUptime`	How long the server has been running.
`iwsInstanceDeathCount`	Number of times server instance processes have gone down.
`iwsInstanceRequests`	Number of requests processed by the server instance.
`iwsInstanceInOctets`	Number of octets received by the server instance. Will show 0 if information is not available.
`iwsInstanceOutOctets`	Number of octets transmitted by the server instance. Will show 0 if information is not available.
`iwsInstanceCount2xx`	Number of 200-level (Successful) responses issued by the server instance.
`iwsInstanceCount3xx`	Number of 300-level (Redirection) responses issued by the server instance.
`iwsInstanceCount4xx`	Number of 400-level (Client Error) responses issued by the server instance.
`iwsInstanceCount5xx`	Number of 500-level (Server Error) responses issued by the server instance.
`iwsInstanceCountOther`	Number of other (neither 2xx, 3xx, 4xx, nor 5xx) responses issued by the server instance.
`iwsInstanceCount302`	Number of 302 (Moved Temporarily) responses issued by the server instance.

Table 10-1 `iws.mib` managed objects and descriptions *(Continued)*

Managed object	**Description**
`iwsInstanceCount304`	Number of 304 (Not Modified) responses issued by the server instance.
`iwsInstanceCount400`	Number of 400 (Bad Request) responses issued by the server instance.
`iwsInstanceCount401`	Number of 401 (Unauthorized) responses issued by the server instance.
`iwsInstanceCount403`	Number of 403 (Forbidden) responses issued by the server instance.
`iwsInstanceCount404`	Number of 404 (Not Found) responses issued by the server instance.
`iwsVsTable`	iPlanet Web Server virtual servers.
`iwsVsEntry`	iPlanet Web Server virtual server.
`iwsVsIndex`	Virtual server index.
`iwsVsId`	Virtual server identifier.
`iwsVsRequests`	Number of requests processed by the virtual server.
`iwsVsInOctets`	Number of octets received by the virtual server.
`iwsVsOutOctets`	Number of octets transmitted by the virtual server.
`iwsVsCount2xx`	Number of 200-level (Successful) responses issued by the virtual server.
`iwsVsCount3xx`	Number of 300-level (Redirection) responses issued by the virtual server.
`iwsVsCount4xx`	Number of 400-level (Client Error) responses issued by the virtual server.
`iwsVsCount5xx`	Number of 500-level (Server Error) responses issued by the virtual server.
`iwsVsCountOther`	Number of other (neither 2xx, 3xx, 4xx, nor 5xx) responses issued by the virtual server.
`iwsVsCount302`	Number of 302 (Moved Temporarily) responses issued by the virtual server.
`iwsVsCount304`	Number of 304 (Not Modified) responses issued by the virtual server.

Table 10-1 `iws.mib` managed objects and descriptions *(Continued)*

Managed object	Description
`iwsVsCount400`	Number of 400 (Bad Request) responses issued by the virtual server.
`iwsVsCount401`	Number of 401 (Unauthorized) responses issued by the virtual server.
`iwsVsCount403`	Number of 403 (Forbidden) responses issued by the virtual server.
`iwsVsCount404`	Number of 404 (Not Found) responses issued by the virtual server.
`iwsProcessTable`	iPlanet Web Server processes.
`iwsProcessEntry`	iPlanet Web Server process.
`iwsProcessIndex`	Process index.
`iwsProcessId`	Operating system process identifier.
`iwsProcessThreadCount`	Number of request processing threads.
`iwsProcessThreadIdle`	Number of request processing threads currently idle.
`iwsProcessConnectionQueueCount`	Number of connections currently in connection queue.
`iwsProcessConnectionQueuePeak`	Largest number of connections that have been queued simultaneously.
`iwsProcessConnectionQueueMax`	Maximum number of connections allowed in connection queue.
`iwsProcessConnectionQueueTotal`	Number of connections that have been accepted.
`iwsProcessConnectionQueueOverflows`	Number of connections rejected due to connection queue overflow.
`iwsProcessKeepaliveCount`	Number of connections currently in keepalive queue.
`iwsProcessKeepaliveMax`	Maximum number of connections allowed in keepalive queue.
`iwsListenTable`	iPlanet Web Server listen sockets.
`iwsListenEntry`	iPlanet Web Server listen socket.
`iwsListenIndex`	Listen socket index.
`iwsListenId`	Listen socket identifier.

Table 10-1 `iws.mib` managed objects and descriptions *(Continued)*

Managed object	Description
`iwsListenAddress`	Address where socket listens.
`iwsListenPort`	Port where socket listens.
`iwsListenSecurity`	Encryption support.
`iwsThreadPoolTable`	iPlanet Web Server thread pools.
`iwsThreadPoolEntry`	iPlanet Web Server thread pool.
`iwsThreadPoolIndex`	Thread pool index.
`iwsThreadPoolEntry`	Thread pool identifier.
`iwsThreadPoolEntry`	Number of requests queued.
`iwsThreadPoolEntry`	Largest number of requests that have been queued simultaneously.
`iwsThreadPoolEntry`	Maximum number of requests allowed in queue.
`iwsInstanceStatusChange`	An `iwsInstanceStatusChange` trap signifies that `iwsInstanceStatus` has changed.
`iwsVsCount503`	Number of 503 (Unavailable) responses issued.
`iwsInstanceCount503`	Number of 503 (Unavailable) responses issued.

Setting Up SNMP

In general, to use SNMP you must have a master agent and at least one subagent installed and running on a your system. You need to install the master agent before you can enable a subagent.

The procedures for setting up SNMP are different depending upon your system. Table 8.1 provides an overview of procedures you will follow for different situations. The actual procedures are described in detail later in the chapter.

Before you begin, you should verify two things:

- Is your system already running an SNMP agent (an agent native to your operating system)?
- If so, does your native SNMP agent support SMUX communication? (If you’re using the AIX platform, your system supports SMUX.)

See your system documentation for information on how to verify this information.

NOTE After changing SNMP settings in the Administration Server, installing a new server, or deleting an existing server, you must perform the following steps:

- (Windows NT) Restart the Windows SNMP service or reboot the machine.
- (Unix) Restart the SNMP master agent using the Administration Server.

Figure 10-1 Overview of procedures for enabling SNMP master agents and subagents.

If your server meets these conditions....	**...follow these procedures. These are discussed in detail in the following sections.**
No native agent is currently running	1. Start the master agent. 2. Enable the subagent for each server installed on the system.
• Native agent is currently running • No SMUX • No need to continue using native agent	1. Stop the native agent when you install the master agent for your Administration Server. 2. Start the master agent. 3. Enable the subagent for each server installed on the system.
• Native agent is currently running • No SMUX • Needs to continue using native agent	1. Install a proxy SNMP agent. 2. Start the proxy SNMP agent. 3. Restart the native agent using a port number other than the master agent port number. 4. Start the master agent. 5. Enable the subagent for each server installed on the system.

If your server meets these conditions....	...follow these procedures. These are discussed in detail in the following sections.
• Native agent is currently running	1. Reconfigure the SNMP native agent.
• SMUX supported	2. Enable the subagent for each server installed on the system.

Using a Proxy SNMP Agent (Unix/Linux)

You need to use a proxy SNMP agent when you already have a native agent running, and you want to use continue using it concurrently with an iPlanet Web Server master agent. Before you start, be sure to stop the native master agent. (See your system documentation for detailed information.)

NOTE To use a proxy agent, you'll need to install it and then start it. You'll also have to restart the native SNMP master agent using a port number other than the one the iPlanet Web Server master agent is running on.

This section includes the following topics:

- Installing the Proxy SNMP Agent
- Starting the Proxy SNMP Agent
- Restarting the Native SNMP Daemon

Installing the Proxy SNMP Agent

If an SNMP agent is running on your system and you want to continue using the native SNMP daemon, follow the steps in these sections:

1. Install the SNMP master agent. See “Installing the SNMP Master Agent” on page 224.
2. Install and start the proxy SNMP agent and restart the native SNMP daemon. See “Using a Proxy SNMP Agent (Unix/Linux)” on page 222.
3. Start the SNMP master agent. See “Enabling and Starting the SNMP Master Agent” on page 225.
4. Enable the subagent. See “Enabling the Subagent” on page 230.

To install the SNMP proxy agent, edit the `CONFIG` file (you can give this file a different name), located in `plugins/snmp/sagt` in the server root directory, so that it includes the port that the SNMP daemon will listen to. It also needs to include the MIB trees and traps that the proxy SNMP agent will forward.

Here is an example of a `CONFIG` file:

```
AGENT AT PORT 1161 WITH COMMUNITY public
SUBTREES 1.3.6.1.2.1.1,
       1.3.6.1.2.1.2,
       1.3.6.1.2.1.3,
       1.3.6.1.2.1.4,
       1.3.6.1.2.1.5,
       1.3.6.1.2.1.6,
       1.3.6.1.2.1.7,
       1.3.6.1.2.1.8
FORWARD ALL TRAPS;
```

Starting the Proxy SNMP Agent

To start the proxy SNMP agent, at the command prompt, enter:

```
# sagt -c CONFIG&
```

Restarting the Native SNMP Daemon

After starting the proxy SNMP agent, you need to restart the native SNMP daemon at the port you specified in the `CONFIG` file. To restart the native SNMP daemon, at the command prompt, enter

```
# snmpd -P port_number
```

where *port_number* is the port number specified in the `CONFIG` file. For example, on the Solaris platform, using the port in the previously mentioned example of a `CONFIG` file, you'd enter:

```
# snmpd -P 1161
```

Reconfiguring the SNMP Native Agent

If your SNMP daemon is running on AIX, it supports SMUX. For this reason, you don't need to install a master agent. However, you do need to change the AIX SNMP daemon configuration.

AIX uses several configuration files to screen its communications. One of them, `snmpd.conf`, needs to be changed so that the SNMP daemon accepts the incoming messages from the SMUX subagent. For more information, see the online manual page for `snmpd.conf`. You need to add a line to define each subagent.

For example, you might add this line to the `snmpd.conf`:

```
smux 1.3.6.1.4.1.1.1450.1 "" IP_address net_mask
```

IP_address is the IP address of the host the subagent is running on, and *net_mask* is the network mask of that host.

NOTE Do not use the loopback address 127.0.0.1; use the real IP address instead.

Installing the SNMP Master Agent

You cannot use the Server Manager to install and start the master SNMP agent unless the server is running as `root`.

To install the master SNMP agent using the Server Manager:

1. Log in as root.
2. Check whether an SNMP daemon (`snmpd`) is running on port 161.

   If no SNMP daemon is running, go to Step 4.

   If an SNMP daemon is running, make sure you know how to restart it and which MIB trees it supports.
3. If an SNMP daemon is running, kill its process.
4. In the Server Manager, choose the SNMP Master Agent Trap page from the Global Settings tab. The Manager Entries page appears.
5. Type the name of the system that is running your network management software.
6. Type the port number at which your network management system listens for traps. (The well-known port is 162.) For more information on traps, see "Configuring Trap Destinations" on page 230.

7. Type the community string you want to use in the trap. For more information on community strings, see “Configuring the Community String” on page 230.
8. Click OK.
9. In the Server Manager, the SNMP Master Agent Community page from the choose Global Settings tab. The Community Strings page appears.
10. Type the community string for the master agent.
11. Choose an operation for the community.
12. Click OK.

Enabling and Starting the SNMP Master Agent

Master agent operation is defined in an agent configuration file named `CONFIG`. You can edit the `CONFIG` file using the Server Manager, or you can edit the file manually. You must install the master SNMP agent before you can enable the SNMP subagent.

If you get a bind error similar to “System Error: Could not bind to port,” when restarting the master agent, use `ps -ef | grep snmp` to check if `magt` is running. If it is running, use the command `kill -9` *`pid`* to end the process. The CGIs for SNMP will then start working again.

This section includes the following topics:

- Starting the Master Agent on Another Port
- Manually Configuring the SNMP Master Agent
- Editing the Master Agent CONFIG File
- Defining sysContact and sysLocation Variables
- Configuring the SNMP Master Agent
- Starting the SNMP Master Agent

Starting the Master Agent on Another Port

The Administration Interface will not start the SNMP master agent on ports other than 161. However, you can manually start the master agent on another port using the following steps:

1. Edit `/`*server_root*`/plugins/snmp/magt/CONFIG` to specify the desired port.
2. Run the start script as follows:

   ```
   cd /server_root/https-admserv

   ./start -shell /server_root/plugins/snmp/magt/magt

   /server_root/plugins/snmp/magt/CONFIG

   /server_root/plugins/snmp/magt/INIT
   ```

The master agent will then start on the desired port. However, the user interface will be able to detect that the master agent is running.

Manually Configuring the SNMP Master Agent

To configure the master SNMP agent manually:

1. Log in as superuser.
2. Check to see if there is an SNMP daemon (`snmpd`) running on port 161.

   If an SNMP daemon is running, make sure you know how to restart it and which MIB trees it supports. Then kill its process.
3. Edit the `CONFIG` file located in `plugins/snmp/magt` in the server root directory.
4. (Optional) Define `sysContact` and `sysLocation` variables in the `CONFIG` file.

Editing the Master Agent CONFIG File

The `CONFIG` file defines the community and the manager that master agent will work with. The manager value should be a valid system name or an IP address.

Here is an example of a basic CONFIG file:

```
COMMUNITY          public
                   ALLOW ALL OPERATIONS

MANAGER            manager_station_name
                   SEND ALL TRAPS TO PORT 162
                    WITH COMMUNITY public
```

Defining sysContact and sysLocation Variables

You can edit the CONFIG file to add initial values for sysContact and sysLocation which specify the sysContact and sysLocation MIB-II variables. The strings for sysContact and sysLocation in this example are enclosed in quotes. Any string that contains spaces, line breaks, tabs, and so on must be in quotes. You can also specify the value in hexadecimal notation.

Here is an example of a CONFIG file with sysContract and sysLocation variables defined:

```
COMMUNITY          public
                   ALLOW ALL OPERATIONS

MANAGER            nms2
                   SEND ALL TRAPS TO PORT 162
                   WITH COMMUNITY public

INITIAL            sysLocation “Server room
501 East Middlefield Road
Mountain View, CA 94043
USA”

INITIAL            sysContact “John Doe
email: jdoe@netscape.com”
```

Configuring the SNMP Master Agent

To configure the SNMP master agent using the Server Manager, perform the following steps:

1. Log in as root.
2. Check whether an SNMP daemon (`snmpd`) is running on port 161.

   If an SNMP daemon is running, make sure you know how to restart it and which MIB trees it supports. Then kill its process.
3. Log in to the Administration Server.
4. From the Administration Server, choose the Global Settings tab and click the SNMP Master Agent Trap link to go to the SNMP Master Agent Control page. The Manager Entries page appears. Complete the following information:

   Manager Station. Enter a valid system name or an IP address for the NMS.

   Trap Port. Enter the port number the NMS uses to listen for traps

   With Community. Enter a community string you want to use for authorization. A common default string is `public`.
5. Click the SNMP Master Community link. The Community Strings page appears. Enter the following community information:

   Community. Specifies the name of the community you want to use for authentication. A common default string is `public`.

   Operation. Specifies the permissions for the new community. Choose from the following:

   - *Allow All Operations*. Allows this community string to request data or reply to messages, and set variable values.
   - *Allow GET Operations*. Allow this community string to only request messages or reply to messages, and not set variables.
   - *Allow ALL Operations*. Allows this community string to only set variable values.

Starting the SNMP Master Agent

Once you have installed the SNMP master agent, you can start it manually or by using the Administration Server.

Manually Starting the SNMP Master Agent

To start the master agent manually, enter the following at the command prompt:

```
# magt CONFIG INIT&
```

The `INIT` file is a nonvolatile file that contains information from the MIB-II system group, including system location and contact information. If `INIT` doesn't already exist, starting the master agent for the first time will create it. An invalid manager name in the `CONFIG` file will cause the master agent start-up to fail.

To start a master agent on a nonstandard port, use one of two methods:

Method one: In the `CONFIG` file, specify a transport mapping for each interface over which the master agent listens for SNMP requests from managers. Transport mappings allow the master agent to accept connections at the standard port and at a nonstandard port. The master agent can also accept SNMP traffic at a nonstandard port. The maximum number of concurrent SNMP is limited by your target system's limits on the number of open sockets or file descriptors per process. Here is an example of a transport mapping entry:

```
TRANSPORT         extraordinary   SNMP
                  OVER UDP SOCKET
                   AT PORT 11161
```

After editing the `CONFIG` file manually, you should start the master agent manually by typing the following at the command prompt:

```
# magt CONFIG INIT&
```

Method two: Edit the `/etc/services` file to allow the master agent to accept connections at the standard port as well as a nonstandard port.

Starting the SNMP Master Agent Using the Administration Server

To start the SNMP master agent using the Administration Server, perform the following steps:

1. Log in to the Administration Server.
2. In the Server Manager, choose the SNMP Master Agent Control page from the Global Settings tab. The SNMP Master Agent Control page appears.
3. Click Start.

   You can also stop and restart the SNMP master agent from the SNMP Master Agent Control page.

Configuring the SNMP Master Agent

Once you've enabled the master agent and enabled a subagent on a host computer, you need to configure the host's Administration Server. This entails specifying community strings and trap destinations.

Configuring the Community String

A community string is a text string that an SNMP agent uses for authorization. This means that a network management station would send a community string with each message it sends to the agent. The agent can then verify whether the network management station is authorized to get information. Community strings are not concealed when sent in SNMP packets; strings are sent in ASCII text.

You can configure the community string for the SNMP master agent from the Community Strings page in the Server Manager. You also define which SNMP-related operations a particular community can perform. From the Server Manager, you can also view, edit, and remove the communities you have already configured.

Configuring Trap Destinations

An SNMP trap is a message the SNMP agent sends to a network management station. For example, an SNMP agent sends a trap when an interface's status has changed from up to down. The SNMP agent must know the address of the network management station so it knows where to send traps. You can configure this trap destination for the SNMP master agent from iPlanet Web Server. You can also view, edit, and remove the trap destinations you have already configured. When you configure trap destinations using iPlanet Web Server, you are actually editing the `CONFIG` file.

Enabling the Subagent

After you have installed the master agent that comes with the Administration Server, you must enable the subagent for your server instance before you attempt to start it. For more information on installing the master agent, see "Installing the SNMP Master Agent" on page 224. You can use the Server Manager to enable the subagent.

To stop the SNMP function on Unix/Linux platforms, you must stop the subagent first, then the master agent. If you stop the master agent first, you may not be able to stop the subagent. If that happens, restart the master agent, stop the subagent, then stop the master agent.

To enable the SNMP subagent, use the SNMP Configuration page in the Server Manager, and start the subagent from the SNMP Subagent Control page. For more information, see the corresponding sections in the online help.

Once you have enabled the subagent, you can start, stop or restart it from the SNMP Subagent Control page or the Services Control Panel for Windows NT.

NOTE After making any SNMP configuration changes, you must click the Apply button, then restart SNMP subagent.

Understanding SNMP Messages

GET and SET are two types of messages defined by SNMP. GET and SET messages are sent by a network management station (NMS) to a master agent. You can use one or the other, or both with the Administration Server.

SNMP exchanges network information in the form of protocol data units (PDUs). These units contain information about variables stored on the managed device, such as the web server. These variables, also known as managed objects, have values and titles that are reported to the NMS as necessary. Protocol data units sent by the server to the NMS are known as “traps.” The use of GET, SET, and “trap” messages are illustrated in the following examples.

NMS-initiated Communication. The NMS either requests information from the server or changes the value of a variable store in the server’s MIB. For example:

1. The NMS sends a message to the Administration Server master agent. The message might be a request for data (a GET message), or an instruction to set a variable in the MIB (a SET message).
2. The master agent forwards the message to the appropriate subagent.
3. The subagent retrieves the data or changes the variable in the MIB.
4. The subagent reports data or status to the master agent, and then the master agent forwards the message back (a GET message) to the NMS.
5. The NMS displays the data textually or graphically through its network management application.

Server-initiated Communication. The server subagent sends a message or “trap” to the NMS when a significant event has occurred. For example:

1. The subagent informs the master agent that the server has stopped.
2. The master agent sends a message or “trap” reporting the event to the NMS.
3. The NMS displays the information textually or graphically through its network management application.

Chapter 11

Tuning Your Server for Performance

For information on tuning your server for performance, please see the Performance Tuning, Sizing, and Scaling Guide for iPlanet Web Server 6.0 at:

```
http://docs.iplanet.com/docs/manuals/enterprise.html
```

Chapter 12

Using Search

The iPlanet Web Server search function allows you to search the contents and attributes of documents on the server. As the server administrator, you can create a customized text search interface tailored to your user community.

NOTE The Search function is not available on Linux platforms.

This chapter contains the following sections:

- About Search
- Configuring Text Search
- Indexing Your Documents
- Performing a Search: The Basics
- Using the Query Operators
- Customizing the Search Interface

About Search

Server documents can be in a variety of formats, such as HTML, Microsoft Excel, Adobe PDF, and WordPerfect, provided that there is a conversion filter available for a particular file format. With the filters, the server converts the documents into HTML as it indexes them, allowing you to use your web browser to view the documents found for your search. For more information, see “About Collections” on page 245.

Users can search through server documents for a specific word or attribute value, obtaining a set of search results that list all documents that match the query. They can then select a document from the list to browse it in its entirety. This provides easy access to server content.

As the server administrator you can:

- Restrict which users and groups are authorized to use text search
- Determine which documents users and groups can access
- Modify the configuration files that govern how text search operates
- Customize the search query and results pages

To enable searching capability on your server, begin by identifying the special configuration needs of your server, and using the several search configuration windows to input these. Then you need to identify the directory or directories of documents that you want prepared for searching, and index the document information into a searchable database, called a collection. The next several sections discuss the details of configuring search and indexing collections.

Configuring Text Search

You can configure several aspects of the search function for your server:

- Collection-specific
- Applying across all collections

Collection-specific configuration controls how documents are indexed into a particular collection, and must be defined before you create the collection. Other configuring actions can be defined at any time, because they only affect the searches themselves.

Collection-specific configuration actions are as follows:

- Define URL mappings for the document directories to be indexed
- Define the pattern files to display for searches on a particular collection

Configuration actions that affect all collections, are as follows:

- Establish access control for files and directories
- Define any words you want dropped from the search
- Define the search parameters

- Turn the search function off and on
- Restrict the amount of memory available for indexing operations

This section includes the following topics:

- Controlling Search Access
- Mapping URLs
- Eliminating Words from Search
- Turning Search On or Off
- Configuring the Search Parameters
- Configuring Your Search Pattern Files
- Configuring Files Manually

Controlling Search Access

The search function accesses the default ACL database for your server. You can restrict access to the documents and directories on your server by defining explicit access control list (ACL) rules, or you can rely on the default access control definitions. You can add users to your server's access control database through the Administration Server's Users & Groups function. You can allow or restrict access through the Restrict Access interface in the Server Manager. For more information about setting access control, see Chapter 8, "Controlling Access to Your Server.

You can configure your server to check access permissions before searching the collection root or displaying search results by using the Search Configuration interface in the Server Manager, as described in "Configuring the Search Parameters" on page 240. When this option is set, the server challenges the user to identify themselves, and checks a user's access privileges before returning the results of a search query.

Mapping URLs

When users search through a collection's files, the resulting documents use a partial Uniform Resource Identifier (URI), to identify them. This security feature prevents users from knowing the complete physical pathname for a file. A URI is set up by mapping a URL to an additional document directory.

For example, if the path for a file is:

```
server_root/Docs/marketing/bizplans/planB.doc
```

you could prevent users from seeing all but the last directory by defining a URL prefix of `plans` and mapping it to:

server_root/Docs/marketing/bizplans

From then on, users need only enter `/plans/planB.doc` to locate the file. For more information, see Chapter 16, “Content Management.

NOTE By default, URLs that are redirected are always escaped. To prevent this, add `escape="no"`. For example:

```
NameTrans fn="redirect" from="/foobar"
url-prefix="index.html" escape="no"
```

The iPlanet Web Server provides three default mappings:

- `/ (slash)` the primary document directory (sometimes called the document root), which initially maps to `server_root/docs`
- `/help` the directory for most of the help files
- `/search-ui` the directory for most of the search interface files

When creating a collection, you must specify which document directory to index. You can only choose a directory that has a URL mapping, or a subdirectory within such a mapped directory. You can create your own mappings to define specific directories.

To map a URL, perform the following steps:

1. Open the Class Manager and select the server instance from the drop-down list.
2. Choose the Content Mgmt tab.
3. Click the Additional Document Directories link.

   The web server displays the Additional Document Directories page.
4. (Optional) Add another directory by entering one of the following.

- URL prefix.

  For example: plans.
- Absolute physical path of the directory you want the URL mapped to.

  For example:

  `C:/iPlanet/Servers/docs/marketing/plans`

5. Click OK.

6. Click Apply.
7. Edit one of the current additional directories listed by selecting one of the following:
    - Edit
    - Remove
8. If editing, select edit next to the listed directory you wish to change.
9. Enter a new prefix using ASCII format.
10. (Optional) Select a style in the Apply Style drop-down list if you want to apply a style to the directory:

    For more information about styles, seeChapter 17, “Applying Configuration Styles
11. Click OK to add the new document directory.
12. Click Apply.
13. Choose Apply Changes to hard start /restart your server.

NOTE Once you create a collection based on an additional document directory, you cannot change the URL mapping or the collection’s entries will target the URL mapping to the wrong physical file location.

Eliminating Words from Search

You can specify words the search engine should not index or search against. These are typically referred to as stop words or drop words**,** and include articles, conjunctions, and prepositions such as *at*, *and*, *be*, *for*, and *the*.

To specify stop words, edit the file named `style.stp.` This file resides in each of the subdirectories `html`, `pdf`, `mail`, and `news` for each collection type in the directory *`server_root`*`\plugins\search\common\style`. Each `style.stp` file controls stop words for that collection type. For example, the style`.stp` file in *`server_root`*`\plugins\search\common\style\html` controls stop words for HTML files in that collection.

Add the stop words to `style.stp`, one per line and left justified. You can use operators such as square brackets ([ ]) to indicate character classes, periods (.) to indicate any character, and plus notation (+) to indicate repeats. For example, the `style.stp` file might contain the following lines:

```
........................................+
at
and
be
[0-9a-zA-Z]
[0-9][0-9][0-9][0-9]+
```

In this example, the first line of periods (in the file by default) indicates that words with 40 or more characters are not to be indexed, as well as the words *at*, *and*, and *be*. [0-9a-zA-Z] indicates that all one letter words are not to be indexed. [0-9][0-9][0-9][0-9]+ indicates that all integers with four or more digits are not to be indexed.

The words you specify are case sensitive, so you need to enter all the case variations of a word. For instance, for *the* you should enter the, THE, and The.

If you want to use stop words, make sure you create the `style.stp` file before you create a collection. Changing the `style.stp` file after a collection has been created, requires you to:

1. Delete the current collection.
2. Change the stop list for the collection type.
3. Recreate the collection.
4. Reindex all the documents in the collection.

Turning Search On or Off

Before users can search your server or web site, you must turn search on. The default setting is for search to be turned off. Turn the search function on or off using the Search State interface in the Server Manager.

Turning search off for a server where users do not use search can improve server performance. You may also want to turn off the search function when you know the server will have heavy traffic, and back on when traffic is lighter. When search is turned off, the search plug-in is not loaded when the HTTP server starts up.

Configuring the Search Parameters

As server administrator, you can set the default parameters that govern what users see when they get search results.

To configure search parameters, perform the following steps:

1. Access the Server Manager and choose Search.
2. Click the Search Configuration link.

   The web server displays the Search Configuration page.
3. Enter the default maximum number of search result items displayed to users at a time in the Default Result Set Size field.

   This number cannot be larger than the value for the largest possible result set size, as defined in Step 4. The default is 20.
4. Enter the maximum number of items in a result set in the Largest Possible Result Set Size field.

   The default is 5000. If you enter 250 as the value, and 1000 documents were found matching the search criteria, users would only have access to either the first 250 or the 250 top-ranked documents.
5. Enter the Date/Time string in Posix format.

   This entry defines how search results are displayed to users. Use the symbols listed in .
6. Enter a default HTML title to be used when a title tag has not been included with an HTML document.

   The typical HTML default is (Untitled) and appears in the search results page for HTML files.
7. Choose Yes to check access permissions on collection root before doing a search if you want to restrict access to the collection.
8. Choose Yes to check access permissions on search results if you want access control checks performed on each document in the search results.
9. Click OK to set your new search configuration.
10. Click Apply.
11. Choose Apply Changes to hard start /restart your server.

Table 12-1 Common Posix Date and Time Formats

Format	Displayed result (example)
%a	Abbreviated week day (for example, Wed)
%A	Full week day (for example, Wednesday)

Table 12-1 Common Posix Date and Time Formats

Format	Displayed result (example)
%b	Abbreviated month (for example, Oct)
%B	Full month (for example, October)
%c	Date and time formatted for current locale
%d	Day of the month as a decimal number (for example, 01-31)
%H	Hour as a decimal number, 24 hour military format (for example, 00-23)
%m	Month as a decimal number (for example, 01-12)
%M	Minute as a decimal number (for example, 00-59)
%x	Date
%X	Time
%y	Year without century (for example, 00-99)
%Y	Year with century (for example, 1999)

Configuring Your Search Pattern Files

Pattern files are HTML files that define the layout of the text search interface. You can associate a pattern file with a search function, and a set of pattern variables to create a specific portion of the interface. In the pattern file you define the look, feel, and function of the text search interface. Pattern files use pattern variables that allow you to customize background color, help text, banners, and so on. In some cases, the values are pathnames to files containing the actual text and graphics that these variables represent; in other cases, the values represent text and HTML.

You can use the default pattern files, or you can create your own customized set of files. The Default start and end pattern files will be used if no start and end pattern files are listed for a collection, or in a multi-collection search. For more information about how to change the user interface, see “Customizing the Search Interface” on page 271.

To define where the search function looks for default pattern files associated with a particular search request, you have to specify paths for the files.

To configure pattern files, perform the following steps:

1. Access the Server Manager and choose Search.
2. Click the Search Pattern Files link.

   The web server displays the Search Pattern Files page.

3. Enter the absolute path for the directory where your pattern files are stored.

   The default start (header), end (footer), and query page pattern files are located in this directory.

4. Enter the relative pathname for the Default Start Pattern File.

   This entry defines the top of the search results page when a collection has no defined header file, or when more than one collection is being searched.

5. Enter the relative pathname for the Default End Pattern File.

   This entry defines the footer of the search results page when for a collection has no defined footer file, or when more than one collection is being searched.

6. Enter the relative pathname for the Pattern File for Query Page.

   This entry defines the search query page that appears when the search function is started.

7. Click OK to configure your search pattern files.

8. Click Apply.

9. Choose Apply Changes to hard start /restart your server.

Configuring Files Manually

The search function examines several configuration files to determine how search is configured on your server. These files define system settings, user-defined variables, and information about your search collections. You normally change this information through the iPlanet Web Server’s Search pages, but you can also modify the files manually with your own text editor. Some of the implications of changing the configuration files in order to customize the user interface are discussed in “Customizing the Search Interface” on page 271.

NOTE Manual modifications to your configuration files are not recommended. If you do make manual modifications, remember to restart the server for your modifications to take effect.

The Configuration Files

The configuration files that govern searching are described in the following list:

- `userdefs.ini`—This user definitions file defines the user-defined pattern variables. It maps to the `userdefs.ini` file for your language (English, German, Japanese, and so on).

  You can customize a search interface for all your pattern files by creating and defining your own pattern variables in the `userdefs.ini` file. For more information, see “User-defined Pattern Variables” on page 277.

- `dblist.ini`—This collection contents file describes collection-specific information. When you create and maintain collections, the `dblist.ini` file is updated for you with information about your collections.

Adjusting the Maximum Number of Attributes

Collections have different sets of default attributes depending on their file format. For example, HTML files have `Title` and `SourceType` attribures. You can also define META-tagged HTML attributes in your HTML files. Some file formats, such as PDF, have a great many default attributes. For more information about the attributes for each format, see “About Collection Attributes” on page 246, and .

Restricting Memory for Indexing

You can set a limit on the amount of RAM available for indexing operations. To do this, manually edit the `[NS-loader]` file to add a line defining a maximum memory amount. For example:

```
NS-max-memory = 32000000
```

The server default uses all of the available system memory. Typically you need to limit the RAM used for indexing if:

- The server is installed on a machine that has less than the suggested minimum RAM
- Server administrators on Windows NT servers require a great deal of indexing, but need memory for other server operations

Restricting Your Index File Size

You can limit how much disk space an index file can consume. To do this, you need to manually edit the `[NS-loader]` file to define a maximum index file size. For example,

```
NS-max-idx-file-size = 1500000
```

An indexing operation typically requires approximately 1.5MB per file, and since there are two files, one of which is temporary, you may need as much as 3MB of disk space for indexing. Setting the file size to 1.5MB per file puts a cap on how large each file can become.

Indexing Your Documents

A database of searchable data is required for users to search. You must create a database, called a collection, that indexes and stores information about the documents, such as their content and file properties.

Searches require collections of files to target. Once the documents are indexed, their contents and file properties, such as their titles, creation dates, and authors, are available for searching.

You can add or delete documents from a collection: optimizing, updating, and managing your collections as needed.

This section includes the following topics:

- About Collections
- About Collection Attributes
- Creating a New Collection
- Configuring a Collection
- Updating a Collection
- Maintaining a Collection
- Scheduling Regular Maintenance
- Removing Scheduled Collection Maintenance

About Collections

When your server administrator indexes all or some of a server’s documents, information about the documents is stored in a collection. Collections contain such information as:

- Format of the documents
- Language they are in
- Searchable attributes

- Number of documents in the collection
- Collection's status
- Brief description of the collection.

For more details, see "Displaying Collection Contents" on page 263.

When creating a collection, you must define the type of files that it contains:

- HTML
- ASCII
- News
- Email
- PDF

During indexing, this definition determines which attributes are indexed, and whether any file conversion is needed.

You can index all the files in a directory, or only those with a specific extension, for example HTML, or PDF.

A collection has records with information about each document that has been indexed. If the document is deleted from the collection, only the collection's entry for that document is removed. The original document is not deleted.

When you have multiple server instances, a collection is only associated with the server instance where it was created. Users can only search collections within that server instance.

About Collection Attributes

Certain file formats have a default set of attributes indexed for files of that type, as shown in Table 12-2.

Table 12-2 The Default Attributes Indexed for Each File Format

File format	Attribute	Type	Description
ASCII	`(none)`	-	-
HTML	`Title`	text	The user-defined title of the file.
	`SourceType`	text	The original format of the document.
NEWS	`From`	text	The source userID of the news item.

Table 12-2 The Default Attributes Indexed for Each File Format *(Continued)*

	`Subject`	text	The text from the subject field of the news item.
	`Keywords`	text	Any keywords defined for the news item
	`Date`	date	The date the news item was created.
EMAIL	`From`	text	The source userID of the email.
	`To`	text	The destination userID of the email.
	`Subject`	text	The text from the email's subject field.
	`Date`	date	The date the email was created.
PDF	`InstanceID`	text	An internal ID number.
	`PermanentID`	text	An internal ID number.
	`NumPages`	integer	The number of pages in the document.
	`DirID`	text	The directory where the PDF file exists.
	`FTS_ModificationDate`	date	The document's last modification date.
	`FTS_CreationDate`	date	The document's creation date.
	`WXEVersion`	integer	The version of Adobe Word Finder used to extract the text from the PDF document.
	`FileName`	text	The Adobe filename specification.
	`FTS_Title`	text	The document's title.
	`FTS_Subject`	text	The document's subject.
	`FTS_Author`	text	The document's author.
	`FTS_Creator`	text	The document's creator.
	`FTS_Producer`	text	The document's producer.
	`FTS_Keywords`	text	The document's keywords.
	`PageMap`	text	The page map, describing the word instances for the page.

By default, HTML collections have `Title` and `SourceType` attributes, but they can be indexed to permit searching and sorting by up to 30 file attributes tagged with the HTML `<META>` tag. You can change the maximum settings for file attributes as discussed in "Adjusting the Maximum Number of Attributes" on page 244.

For example, a document could have these lines of HTML code:

```
<META NAME="Writer" CONTENT="R. Hunter">
<META NAME="Song" CONTENT="Stella Blue">
```

If this document was indexed with its META tags extracted, you could search it for specific values in the writer or product fields. For example, you could enter this query: `Writer <contains> Hunter` or `Song <contains> Blue`.

NOTE Attribute values in META-tagged fields are text strings only, which means that all numeric values, such as date and time, are sorted as text. Any illegal HTML characters in a META-tagged attribute are replaced with a hyphen.

Creating a New Collection

You can only have twelve collections on your server. To use a thirteenth collection, you must first remove an existing collections using Search /Maintain Collection.

You can only have entries for a maximum of 16 million documents in your collections. A document that is indexed in multiple collections counts as multiple documents. It is best to create new collections of over 10,000 documents at low-traffic times, or the indexing operation may affect your system's performance.

You can create a collection that indexes the content of all or some of the files in a directory. You can define collections that contain only one kind of file, or you can create a collection of documents in various formats that are automatically converted to HTML during indexing. When you define a multiple format collection with the auto-convert option the indexer first converts the documents into HTML, and then indexes their contents. The converted HTML documents are put into the `html_doc` directory in the server's search collections folder.

The file format you choose defines which default attributes are used in the collection, and whether automatic HTML conversion of the content is needed during indexing. For information about the attributes for each format, see , and "About Collection Attributes" on page 246.

Regardless of the file type chosen, the content of the file is always indexed. If you choose HTML as the file type, the server creates the collection with the HTML default attributes, and does not attempt to convert any non-HTML files you try to index. If you index HTML files into an ASCII collection, even the HTML markup tags are indexed as part of the file's contents, and the contents are displayed as raw text.

NOTE You need to have at least 3MB of available disk space on your system to create a collection. For information on how you can restrict the size of the index files, see “Restricting Your Index File Size” on page 244.

To create a new collection, perform the following steps:

1. Access the Server Manager and select the server to create a collection for form the drop-down list.
2. Choose the Search tab.
3. Click the New Collection link.

   The web server displays the Create a Collection page.

   Select:

   - The current document directory from the The Directory to Index field
   - A different document directory defined for the server drop-down list
   - View for a list of files and subdirectories

   For more information about additional document directories, see “Mapping URLs” on page 237.
4. Accept the default *.html for the Documents Matching field, or define your own wildcard expression.

   You can define multiple wildcards in an expression. For example:

   ```
   (*.htm|*.html or *(.htm|.html)
   ```

   For details of the syntax for wildcard patterns, see “Using Wildcards” on page 270.

NOTE You cannot index a file that includes a semi-colon (;) in its name. Rename these files.

5. Choose Yes to include subdirectories within the specified directory in the index.
6. Enter a name for your collection in the Collection Name field.

   The collection name is used for collection maintenance. This is the physical file name for the file, so follow the standard directory-naming conventions for your operating system. You can use up to a maximum of 128 characters. Spaces are converted to underscores.

NOTE Do not use accented characters in the collection name. If you need accented characters, exclude the accents from the collection name, but use them in the label. The label is displayed to the user from the search interface.

7. (Optional) Enter a user-defined name for your collection in the Collection Label field.

   This name is displayed when users perform a text search. Make your collection's label as descriptive and relevant as possible. You can use any characters except single or double quotation marks, up to a maximum of 128 characters.

8. (Optional) Enter a description for your collection (up to a maximum of 1024 characters) in the optional Description field.

   This description is displayed in the collection contents page.

9. Select the type of files the collection is to contain:

   ❍ ASCII

   ❍ HTML

   ❍ News

   ❍ Email

   ❍ PDF

10. Select whether or not to extract META-tagged attributes from HTML files during indexing.

    Only select this option for HTML collections. Extracting these attributes allows you to search their values. You can index a maximum of 30 different user-defined META tags in a document.

11. Choose the collection's language from the drop-down list.

    The default is English, labeled "English (ISO-8859-1)." For more information on character sets, see Chapter 16, "Content Management.

12. Click OK to create a new collection.

NOTE Once you begin indexing a collection, you cannot stop the process until either the indexing is complete, or you reboot the system. Shutting down your server does not kill the process.

13. Click Apply.

14. Choose Apply Changes to hard start /restart your server.

Configuring a Collection

After you have created a collection you can modify some of its initial settings. These settings reside in the collection information file, `dblist.ini`. When you reconfigure a collection the `dblist.ini` file is updated to reflect your changes. For more information about the configuration files, see “Configuring Files Manually” on page 243. You can modify your settings to:

- Revise the description
- Change the label
- Define a different URL for the documents
- Define how to highlight displayed documents
- Define which pattern files to use
- Define how to format dates

You should avoid making unnecessary changes to a collection’s settings.

To configure a collection, perform the following steps:

1. Access the Server Manager and select the server instance from the drop-down list that the collection is in.

2. Choose the Search tab.

3. Click the Configure Collection link.

   The web server displays the Configure Collection page.

4. Choose the collection to configure.

5. You can enter or change:

    ❍ A description using up to 1024 characters in the optional Description field.

    ❍ A user-defined name in the Label field.

    ❍ A URL in the URL for Documents field, if that has changed.

      For example, you might change the URL mapping from `publisher/help`, to the simpler `/helpFiles`.

    ❍ The HTML tagging the server will use when highlighting a search query word or phrase in a displayed document in the Highlight Begin and Highlight End fields.

   The default is bold, with the `<b>` and `</b>` tags, but you can add to this or change it. For example, you could add `<blink><FONT COLOR = #FF0000>` and the corresponding `</blink></FONT>` to highlight with blinking bold red text.

6. Select the format for input dates.

7. Define or change the default pattern files for displaying the search results for:

    ❍ Header

    ❍ Footer

    ❍ Record

8. Enter or change the name of the pattern file displaying a single highlighted document from the list of search results in the Result Pattern File field.

9. Click OK to change the collection configuration.

10. When the server finishes configuring the collection, click Apply.

11. Choose Apply Changes to hard start /restart your server.

Updating a Collection

After you have created a collection, you may want to add or remove files. When adding documents, the file contents are indexed and converted, if necessary. If you are removing documents, the entries for the files are removed from the collection along with their metadata. The original documents are not affected, only their entries in the collection.

NOTE If you selected the Extract Metatags option when you created a collection, then the META-tagged HTML attributes are indexed whenever you add new documents to it.

To update a collection, perform the following steps:

1. Access the Server Manager and select the server instance from the drop-down list that the collection is in.
2. Choose the Search tab.
3. Click the Update Collection link.

   The web server displays the Update Collection page.
4. Choose the collection to update.

   The list of documents displays which have index entries in the currently selected collection. Each list holds 100 records; use the Prev and Next buttons to display more lists for collections greater than 100 files.
5. Enter a single filename, or use wildcards to specify the type of files you want added or removed from the collection in the Documents Matching field.

   Entering a wildcard such as `*.html`, allows only files with this extension to be updated. For files within a subdirectory enter the pathname as it appears in the list of files. For example: `frenchDocs/*.html`

CAUTION Be careful entering wildcard expressions. Entering `index.html` allows you to add or remove the index file from the current collection, but `*/index.html` causes you to add or remove all `index.html` files in the collection.

6. Choose whether to include subdirectories.
7. Click either:
   - AddDocs to add the indicated files and subdirectories
   - RemoveDocs to remove the indicated files
8. Click Apply.
9. Choose Apply Changes to hard start /restart your server.

Maintaining a Collection

Periodically, you may want to maintain your collections. With normal usage, these tasks may not be necessary, unless you index and update collections frequently.

You can perform the following collection management tasks:

- **Optimize collections**—You can optimize a collection to improve performance if you frequently add, delete, or update documents or directories in your collections. An analogy is defragmenting your hard drive. Optimizing is not done automatically; you must manually optimize after you reindex or update a collection. You might want to optimize a collection just before publishing it to another site, or before putting it onto a read-only CD-ROM.
- **Reindex**—You can reindex a collection. Each file that already has an entry in the collection can be located and its attributes and contents reindexed. META-tagged attributes will be extracted if that option was selected when the files were originally indexed into the collection. This does not return to the original criteria for creating the collection, say `*.html`, and add any new documents that fit the original criteria. Collection entries are removed when the source documents have been deleted and can no longer be found.
- **Remove**—You can remove a collection. This only removes the collection, not the original source documents.

NOTE Do not use your local file manager to remove collections. When you try to execute a search before restarting your server again, the search will fail.

To perform any of these collection management tasks, use The Maintain Collection link in the Server Manager.

Scheduling Regular Maintenance

You can schedule collection maintenance at regular intervals, and you can set up separate maintenance schedules for optimizing and reindexing. With normal usage, these tasks may not be necessary, unless you index and update collections frequently. For example, some very active web sites may require frequent reindexing if new documents are added on a daily basis.

A common combination of tasks regularly schedules:

- Cleaning out deleted entries with reindex and update operations
- Adding entries for new documents matching your collection criteria.
- Updating a collection by entering new indexing criteria for the collection

To optimize, reindex, or update your collection, perform the following steps:

1. Choose Search from the Server Manager.
2. Click the Schedule Collection Maintenance link.

   The web server displays the Schedule Collection Maintenance window.
3. Choose a collection from the drop-down list.

   This lists all the collections that you have created.
4. Choose an action from the drop-down list:

   - Reindex
   - Optimize
   - Update

   You can set up different schedules for different actions on the same collection. If you choose to update your collection, two extra fields are displayed for entering the document matching criteria, and for including documents found in subdirectories that match your criteria.
5. Enter the time of day when you want the scheduled maintenance to take place in the Schedule Time field.

   Use a military format (HH:MM). HH must be less than 24 and MM must be less than 60. You must enter a time.
6. Check one or more days in the section labeled Schedule Day(s) of the Week.

   You can select all days, but you must select at least one day.
7. Click OK to schedule the maintenance.

For Unix/Linux users, to make your newly scheduled maintenance take effect, you must restart the `ns-cron` process from the Administration Server.

To restart the `ns-cron` process, perform the following steps:

1. From the Administration Server, Choose Global Settings.
2. Click the Cron Control link.
3. If `ns-cron` is already on, click Restart to restart it. If `ns-cron` is not on, click Start to start it up.

   In either case, your regularly scheduled maintenance will now be able to take place automatically.

Removing Scheduled Collection Maintenance

You can remove scheduled regular maintenance of a collection if no longer needed.

To unschedule collection maintenance, perform the following steps:

1. Choose Search from the Server Manager.
2. Click the Remove Scheduled Collection Maintenance link.

   The web server displays the Remove Scheduled Collection Maintenance window.
3. Choose a collection from the drop-down list for Choose Collection.

   This lists all your collections for which you have set up regular maintenance.
4. Choose an action from the drop-down list: Reindex, Optimize, or Update.

   In the lower part of the frame, you can see the time and days of the week when the scheduled maintenance is currently scheduled to take place.
5. Click OK to remove the scheduled maintenance.

For Unix/Linux users, to make your newly scheduled maintenance take effect, you must restart the `ns-cron` process.

To restart the `ns-cron` process, perform the following steps:

1. From the Administration Server, choose Global Settings.
2. Click the Cron Control link.
3. If `ns-cron` is already on, click Restart to restart it. If `ns-cron` is not on, click Start to start it up.

   In either case, your regularly scheduled maintenance will no longer take place.

Performing a Search: The Basics

Users are primarily concerned with asking questions of the data in the search collections and getting a list of documents in return. When you install the iPlanet Web Server a default set of search query and result forms are included. These allow users a simple method of accessing the search function.

There are four parts to text searching:

- **making a query**—the user enters search criteria
- **displaying search results**—the server displays a list of the documents that match your criteria
- **viewing a document**—the user can view a specific highlighted document from the search results list
- **viewing the contents of a collection**—the user can look at the information that is maintained for each of your collections.

NOTE If the search function is turned off, these query forms are not available.

This section includes the following topics:

- Search Home Page
- A Search Query
- Guided Search
- Advanced Search
- The Search Results
- Displaying Collection Contents

Search Home Page

The search home page (see: `http://`*`server_root`*`:`*`port`*`/search`) provides individual links to each of the three search query interfaces as well as an online QuickStart tutorial on customizing the interface. The tutorial discusses the various pattern files and gives examples of how they can be changed to produce different results.

A Search Query

The default installation of iPlanet Web Server includes three search query pages: standard and advanced HTML queries and a Java-based guided query.

With the standard search query you select a collection to search against, and enter in a word or phrase to search for using the query language operators.

With the guided Java-based search interface you can use the many drop-down lists to easily construct a query. To do this Java must be enabled for your browser.

With the advanced HTML page, you have the additional options of selecting multiple collections to search through, establishing a sort sequence for the results, and defining how many documents are to be displayed on a page. Typically, clicking the Prev and Next arrows moves you through the pages of results.

To perform a standard search, perform the following steps:

1. Enter the following URL in the location field in your web browser:

   `http://`*server_root:port*`/search`

2. In the search query page that appears, choose the collection you want to search through from the drop-down list in the Search In field.

3. Enter the word or phrase for your search query in the For field. You can create complex queries by combining operators. For details about the search operators, see “Using the Query Operators” on page 263.

4. Click the Search button to execute your query.

Guided Search

You can choose to use the Java-based guided search interface, which helps you construct the query. This is especially useful if you want to build a query that has several parts, say searching for a word in the documents’ content as well as a specific attribute value.

Make sure Java is enabled for your browser. Use the Preferences /Languages to enable.

NOTE The attributes for Version Control and Link Management are no longer used in iPlanet Web Server. However, note that if you perform a guided search, iPlanet Web Server may still return them; consequently, do not use these variables.

There are two ways to obtain the guided search page:

- Through the Search home page

- Through the standard search query page

To access the guided search interface through the Search home page, perform the following steps:

1. Enter the following URL in the location field in your web browser:

   `http://*server_root:port/*search`

2. Click the **G**uided Search link on the home page.

To access guided search through the standard search query page, perform the following steps:

1. Go to the standard search query page by typing the following URL in the location field in your web browser:

   `http://*server_root:port/*search`

2. Click Guided Search on the standard search page and the guided Java-based query page is displayed.
3. Choose the collection you want to search through from the drop-down list in the Search In field.
4. Use the For drop-down list to select the type of element you wish to search for. In this example, choose Words.
5. Enter in the word you want to search for in the blank text field.

   For details about the search operator, see “Using the Query Operators” on page 263.

6. Click Add Line to add the first part of the query. The word appears in the large text display box at the bottom of the form.
7. Choose another element from the drop-down list to add to your query. In this example, choose Attribute.
8. Choose the attribute you want to search against from the new drop-down list of all attributes that are available for the chosen collection.
9. Choose a query operator (Contains, Starts, Ends, Matches, Has a substring), or logical operator (=, <, , <=, =) for your query from the drop-down list above the text input field.
10. Enter the attribute value you want to search for in the blank text field.

11. Choose:
    - Add Line to add another line for yor query
    - Undo Line to remove the last line you added
    - Clear to remove the entire query
12. Click the Search button to execute the search.

Advanced Search

You can choose to use the advanced HTML search interface, which helps you construct the query. This is especially useful if you want to create a query that searches through more than one collection, or that produces results sorted by a specific attribute value.

There are two ways to obtain the advanced HTML search page:

- Through the Search home page
- Through the standard search query page

To access advanced HTML through the Search home page, perform the following steps:

1. Enter the following URL in the location field in your web browser:

   `http://server_root:port/search`

2. Click the Advanced HTML Search link on the home page.

To access advanced HTML search through the standard search query page, perform the following steps:

1. Go to the standard search query page by typing the following URL in the location field in your web browser:

   `http://server_root:port/search`

2. Disable Java for your browser using Preferences /Languages.
3. Click Guided Search on the standard search page and the web server displays the advanced HTML query page.
4. Enter the word or phrase you want to search for in the For field.

   You can create complex queries by combining operators. For details about the search operators, see “Using the Query Operators” on page 263.

5. Enter in one or more attributes to sort the results by.

   The default is an ascending sort order, but you can indicate a descending sort order with a minus. For more information about sorting, see “Sorting the Results” on page 262.

6. Expand or limit the number of matching documents you want the search to return depending on how many fields are listed for each document in the search results page, or how many you want to see at a time.

   The Prev and Next buttons allow you access to additional pages of documents if there are too many to fit on a page at once.

7. Use the drop-down list in the Search In field to choose the collection you want to search through.

   You can select more than one collection by holding down the Ctrl key as you click on another collection. All collections in a query must be in the same language.

8. Click the Search button to execute your query.

The Search Results

There are two standard types of search results:

- A list of all documents that match the search criteria
- The text of a single document that you selected from the list of matching documents

Your access permissions are checked at several points during the search process:

- When a user clicks on the icon displayed for a document in the search results which displays the highlighted version
- When searching on a collection that has the option NS-collection-acl-check set to yes. NS-collection-acl-check applies to all collections. When it is set, ACLs that are set on URIs matching the primary document directory defined for the collections (in `dblist.ini`) will be honored by not allowing search to be done on those collections

Listing Matched Documents

With the default installation of the iPlanet Web Server, when you execute a search from either the simple or advanced search query pages, the server returns a list of the documents that match your search criteria. The list gives some standard information about each file, depending on the collection’s format. For example, the default results page for email collections give subject, to, from, and date for each entry; and news collections give subject, from, and date for each entry.

The file format in the collection indicates which default attributes are available for searching. For information about the attributes for each format, see "About Collection Attributes" on page 246.

For entries resulting from a search that checks for comparative proximity of words to each other, or for the exactness of the match, the file's ranking can be provided by showing a score.

If there are more matching documents than can fit on a page, click Next to see the next batch. You can always execute a new search by entering new query data and clicking Search.

Sorting the Results

By default, or if you don't enter anything in the Sort By field on the advanced HTML query page, all documents matching the search are returned according to:

- Their relevance ranking (for queries that consider this)
- Their position in the server file database (for other queries)

If you enter an attribute name in the Sort By field, the documents are displayed in an ascending sort sequence. You can list the documents in a descending sort sequence by adding a minus sign (-) prefix to the attribute, as in -keywords or -title. You can do a multiple sort, by typing in more than one field, as in `Author,-PubDate`.

In a short query, sort order usually isn't critical, but in queries that result in a great many matches, you may want to set a sort value in order to obtain useful search results. However, using a special sort sequence may impact the search's performance.

NOTE Attribute values in META-tagged fields are text strings only, which means that all numeric values, such as date and time, are sorted as text. Any illegal HTML characters in a META-tagged attribute are replaced with a hyphen.

Displaying a Highlighted Document

In the default installation of iPlanet Web Server, when you obtain a list of the documents that match your search criteria, you can select a single document to view in your web browser. Depending on how the pattern files are set up, the word you entered as your original search query can be highlighted in the displayed document with color, boldface text, or blinking.

To view a highlighted document, click on the document's entry in the search results. The field you use to access the highlighted document depends on how your search interface has been designed; in the default installation you click the icon shown next to the document's listing. Additional code behind the icon's link defines how to format the displayed document with the search query highlighted.

In the default search results page, if you click the file's URL, the file opens in your browser without any special highlighting.

In the case of documents that have been converted into HTML, the URL points you to the original document. To get to the converted HTML document, click the document's title.

Displaying Collection Contents

You can display the contents of your collection database to see which attributes are set for each collection. The default installation of iPlanet Web Server uses the `HTML-description.pat` file to display information about each of your displayable (`NS-display-select = YES`) collections in the `dblist.ini` file. The collection contents typically include these items:

- Collection name, label, and description
- Collection format
- Number of attributes in the collection and a list of their names
- Number of documents in the collection
- Collection size and status
- Language and character set
- Input and output date formats

To display your collection database contents, use the following URL:

```
http://server_root:port/search?NS-search-page=c
```

Using the Query Operators

To perform an effective search, you need to know how to use the query operators. You can only do Boolean searches, so all the subsequent information is based on Boolean search rules.

NOTE The query language is not case-sensitive. The examples use uppercase for clarity only.

The search engine interprets the search query based on a set of syntax rules. For example, by entering the word *region*, the actual word region and all its stemmed variations, such as *regions* and *regional*, are found. The search results are ranked for importance, meaning how close the matched word comes to the originally input search criteria. In the example above, *region* would rank higher than any of the stemmed variants.

Not all queries rank their results. Only those queries that can have varying degrees of matching can be ranked. For example, <CONTAINS queries either do or do not contain the given string, but <NEAR queries can be ranked according to how close the words are to each other. Words closer together are listed at the top of the search results, while those that are far apart are put at the bottom of the results.

This section includes the following topics:

- Default Assumptions
- Search Rules
- Determining Which Operators To Use
- Using Wildcards

Default Assumptions

The search query language has some implicit defaults and assumptions that dictate how your input is interpreted. In some cases, you can circumvent the defaults, but the search engine decides what results to return using:

- **<STEM>** Search finds all documents that contain any stemmed variant of the search word or phrase. The search engine looks at the meaning of the word, not just its spelling. For example, if you want to search plan, the results would include documents that contain planning and plans, but not those that contain plane or planet.
- **<MANY>** Search considers how often the search word or phrase appear in the found documents and ranks the results for frequency or relevancy.

- **<PHRASE>** Search considers words separated by spaces to be part of a phrase. For example, Monterey otter is interpreted as a phrase, and both words must be present and together to be found. Such a search would not find documents containing sea otter or Monterey Bay.

  In any case where it's not clear that two words are to be considered as a phrase, you can use parentheses for clarity. For example, <PHRASE> (rise "and" fall).
- **OR** Search considers each word or phrase in the query separated by a comma to be optional, although at least one must be present. In effect, this is an implicit OR operation. For example, *Monterey, otter* is interpreted as find documents that contain either *Monterey* or *otter*. Note that angle brackets are not required for `OR`.

Search Rules

To create complex searches, you can:

- Combine query operators
- Manipulate the query syntax
- Include wildcard characters

Angle Brackets

With the exception of the `AND`, `OR`, `NOT`, and the date and numeric comparison operators, you need to enclose query operators in angle brackets, as in `<CONTAINS>` and `<WILDCARD>`.

Combining Operators

You can combine several query operators into a single query to obtain precise results. For example, you can input the following query to limit your search to those documents that have *Bay and Monterey,* but excludes those that also mention *Aquarium:*

```
Monterey AND Bay NOT <CONTAINS> Aquarium
```

You can achieve even greater precision by including some implicit phrases, as in the following query that finds documents that refer to the *Monterey Bay Aquarium* by its full name and also mention *otters* but do not refer to *shark:*

```
Monterey Bay Aquarium AND otter AND NOT shark
```

Using Query Operators as Search Words

You can use any of the query operators as a search word, but you must enclose the word in quotation marks. For example, you could search for documents about the *ebb and flow* of the tides with the following query:

```
<CONTAINS> ebb "and" flow
```

Canceling Stemming

You can cancel the implicit stemming by using quotation marks around a word. For example, you can be exact by using a query such as this:

```
"plan"
```

This search only results in documents that contain the exact word *plan*. It ignores documents with *plans* or *planning*.

Modifying Operators

You can use AND, OR, and NOT to modify other operators. For example, you may want to exclude documents with titles that contain the phrase *theme park*. A query such as this would solve this problem:

```
Title NOT <CONTAINS> theme park
```

Determining Which Operators To Use

Use the following reference to help determine which operators to use. Note that the query language is not case-sensitive, so `<starts>` and `<STARTS>` are equivalent. This document uses uppercase for clarity only.

Table 12-3 Deciding which operator to use

Type of Search	Valid Operators	Examples
Finding documents by date or numeric value comparison.	• greater than (>) • greater than or equal to (>=) • less than (<) • less than or equal to (<=)	`DATE >= 06-30-96` Finds documents created on or after June 30, 1996.

Table 12-3 Deciding which operator to use

Type of Search	Valid Operators	Examples
Finding words or phrases in specific document fields or in specific locations in the field.	• <STARTS> • <CONTAINS> • <ENDS> • is equal to (=)	`Title <STARTS> Help` Finds documents with titles that start with *Help*.
Finding two or more words in a document.	• AND • <NEAR/1>	`specifications AND review` Finds documents that contain both *specifications* and *review*.

The following table describes some commonly used operators and provides examples of how to use each one. All are relevance ranked except where explicitly noted.

Table 12-4 Query language operators

Operator	Description	Examples
`AND`	• Adds mandatory criteria to the search. • Finds documents that have all of the specified words.	`Antarctica AND mountain climb` Finds only documents containing both *Antarctica* and *mountain climb* plus all the stemmed variants, such as *mountain climbing*.
`<CONTAINS>`	• Finds documents containing the specified words in a document field. The words must be in the exact same sequential and contiguous order. • You can use wildcards. Only alphanumeric values. • Does not rank documents for relevance.	`Title <CONTAINS> higher profit` Finds documents containing the phrase *higher profit* in the title. Ignores documents with *profits higher* in the title.
`<ENDS>`	• Finds documents in which a document field ends with a certain string of characters. • Does not rank documents for relevance.	`Title <ENDS> draft` Finds documents with titles ending in *draft*.
equals (=)	• Finds documents in which a document field matches a specific date or numeric value	`Created = 6-30-96` Finds documents created on June 30, 1996.

Table 12-4 Query language operators *(Continued)*

Operator	Description	Examples
greater than (>)	• Finds documents in which a document field is greater than a specific date or numeric value.	`Created > 6-30-96` Finds documents created after June 30, 1996.
greater than or equal to (>=)	• Finds documents in which a document field is greater than or equal to a specific date or numeric value.	`Created >= 6-30-96` Finds documents created on or after June 30, 1996.
less than (<)	• Finds documents in which a document field is less than a specific date or numeric value.	`Created < 6-30-96` Finds documents created before June 30, 1996.
less than or equal to (<=)	• Finds documents in which a document field is less than or equal to a specific date or numeric value.	`Created <= 6-30-96` Finds documents created on or before June 30, 1996.
`<MATCHES>`	• Finds documents in which a string in a document field matches the character string you specify. • Ignores documents that contain partial matches. • Does not rank documents for relevance.	`<MATCHES> employee` Finds documents containing *employee* or any of its stemmed variants such as *employees*.
`<NEAR>`	• Finds documents that contain the specified words. The closer the terms are to each other in the document, the higher the document's score.	`stock <NEAR> purchase` Finds any document containing both *stock* and *purchase*, but gives a higher score to a document that has *stock purchase* than to one that has *purchase supplies and stock up*.
`<NEAR/N>`	• Finds documents in which two or more specified words are within N number of words from each other. N can be an integer up to 1000. Also ranks the documents for relevance based on the words' proximity to each other.	`stock <NEAR/1> purchase` • Finds documents containing the phrases *stock purchase* and *purchase stock.* • Ignores documents containing phrases like *purchase supplies and stock up* because *stock* and *purchase* do not appear next to each other. • When N is 2 or greater, finds documents that contain the words within the range and gives a higher score for documents which have the words closer together.

Table 12-4 Query language operators *(Continued)*

Operator	Description	Examples
`NOT`	• Finds documents that do not contain a specific word or phrase. **Note:** You can use `NOT` to modify the `OR` or the `AND` operator.	`surf AND NOT beach` Finds documents containing the word *surf* but not the word *beach.*
`OR`	• Adds optional criteria to the search. • Finds any document that contains at least one of the search values.	`apples OR oranges` Finds documents containing either *apples* or *oranges*.
`<PHRASE>`	• Finds documents that contain the specified phrase.A phrase is a grouping of two or more words that occur in a specific order.	`<PHRASE> (rise "and" fall)` Finds documents that include the entire phrase *rise and fall*. The *and* is in quotes to force the search to interpret it as a literal, not as an operator.
`<STARTS>`	• Finds documents in which a document field starts with a certain string of characters. • Does not rank documents for relevance.	`Title <STARTS> Corp` Finds documents with titles starting with *Corp*, such as *Corporate* and *Corporation*.
`<STEM>` (English only)	Finds documents that contain the specified word and its variants.	`<STEM> plan` Finds documents that contain *plan, plans, planned*, *planning*, and other variants with the same meaning stem. Ignores similarly spelled words such as *planet* and *plane* that don’t come from the same stem.
`<SUBSTRING>`	• Finds documents in which part or all of a string in a document field matches the character string you specify. • Similar to `<MATCHES>`, but can match on a partial string. • Does not work with wildcards. • Does not rank documents for relevance.	`<SUBSTRING> employ` Finds documents that can match on all or part of *employ,* so it can succeed with *ploy*.

Table 12-4 Query language operators *(Continued)*

Operator	Description	Examples
`<WILDCARD>`	• Finds documents that contain the wildcard characters in the search string. You can use this to get words that have some similar spellings but which would not be found by stemming the word. • Some characters, such as * and ?, automatically indicate a wildcard-based search, so you don't have to include the word `<WILDCARD>`.	<WILDCARD> plan* • Finds documents that contain *plan*, *plane*, and *planet* as well as any word that begins with *plan*, such as *planned*, *plans*, and *planetopolis*. • See the next section for more details and examples.
`<WORD>`	Finds documents that contain the specified word.	`<WORD> theme` Finds documents that contain *theme*, *thematic*, *themes*, and other words that stem from *theme*.

Using Wildcards

You can use wildcards to obtain special results. For example, you can find documents that contain words that have similar spellings but are not stemmed variants. For example, *plan* stems into *plans* and *planning*, but not *plane* or *planet*. With wildcards, you can find all of these words.

Only the * and ? wildcard charcters are supported. They automatically indicate a wildcard-based search, and do not require you to use the `<WILDCARD>` operator as part of the expression.

Table 12-5 Wildcard Operators

Character	Description
*	• Specifies 0 or more alphanumeric characters. For example, `air*` finds documents that contain *air*, *airline*, and *airhead*. • Cannot use this wildcard as the first character in an expression. • This wildcard is ignored in a set of ([ ]) or in an alternative pattern ({ }). • With this wildcard, the`<WILDCARD>` operator is implicit.

Table 12-5 Wildcard Operators *(Continued)*

Character	Description
?	• Specifies a single alphanumeric character, although you can use more than one ? to indicate multiple characters. For example, ?at finds documents that contain *cat* and *hat*, while ??at finds documents that contain *that* and *chat*. • This wildcard is ignored in a set of ([ ]) or in an alternative pattern ({ }). • With this wildcard, the`<WILDCARD>` operator is implicit.

Non-alphanumeric Characters

You can only search for non-alphanumeric characters if the `style.lex` file used to create the collection is set up to recognize them. This file is in the HTML, news, and mail subdirectories of the *server_root*`\plugins\common\` directory.

Customizing the Search Interface

As server administrator, you can customize the search interface to meet specific user requirements. All of the HTML-based forms that the user sees are defined through a set of pattern files to:

- Display formats for the search results page header and footer
- Display each search result record listed in response to a query

There is a set of pattern variables to construct the forms used for search input and output. Many of the variables are defined in the system and user configuration files `userdefs.ini` and `dblist.ini`, which are discussed in "Configuring Files Manually" on page 243.

NOTE	The search home page, at `http://server_root:port/search` also provides an introduction to the search interface, as well as an online QuickStart tutorial on customizing the interface. The tutorial discusses the various pattern files, and gives examples of how they can be changed to produce different results.

This section includes the following topics:

- Dynamically Generated Headers and Footers
- HTML Pattern Files

- Search Function Syntax
- Using Pattern Variables

Dynamically Generated Headers and Footers

You can specify dynamically generated headers and footers. To accomplish this, add the `add-headers` and `add-footers` directives to your obj.conf file as Service functions. These directives require either a path or `URI` parameter. Use the path parameter to specify a static file as the header or footer. For example:

```
Service fn="add-headers" path="/export2/docs/header.html"
```

```
Service fn="add-footer" path="/export2/docs/footer.html"
```

Use the `URI` parameter to specify a dynamically generated file, such as a CGI program, as the header or footer. For example:

```
uri="/cgi-bin/header.cgi"
```

These Service functions should precede the actual Service function that will answer the request, such as `send-file` or `send-cgi`.

HTML Pattern Files

A good place to begin customizing the interface is to modify the existing pattern files. After you understand pattern variables and how they work, you can create your own pattern files, and change the configuration files and other pattern files to point to them. In the default installation of iPlanet Web Server, the pattern files are in this directory: `server_root\plugins\search\ui\text`. It’s a good idea to make copies of your original pattern files so you can restore them afterwards.

There are pattern files for different kinds of collections: email, news, ASCII, PDF, and HTML. There are several general types of pattern files, each of which has a particular use. A file prefix designates which type of file the pattern file is for, for example, `ASCII-record.pat`, or `EMAIL-record.pat`. The following list describes the general pattern file types:

- `NS-query.pat` displays the standard and advanced query pages. Contains HTML calling the Web Search (the “Search the Web” box) as part of the search query page.
- `tocstart.pat` displays the header across the top of the search results page.
- `tocrec.pat` displays each document listed on the search results page.
- `tocend.pat` displays the footer across the bottom of the search results page.

- `record.pat` displays a single highlighted document from the search results page (for more information, see “Displaying a Highlighted Document” on page 262).
- `descriptions.pat` displays the collection contents.

The pattern files contain HTML formatting instructions, which define how elements look; and HTML search arguments and variables, which define the text label or value that is displayed.

There are three kinds of pattern variables (discussed further in “Using Pattern Variables” on page 276):

- **User defined**, in the `userdefs.ini` file, with a `$$` prefix (see “User-defined Pattern Variables” on page 277).
- **defined in the configuration files**, `dblist.ini` files, with a `$$NS-` prefix (for more information, see “Configuration File Variables” on page 279).
- **search macros and variables generated by a pattern file**, with a `$$NS-` prefix (for more information, see “Macros and Generated Pattern Variables” on page 281).

The following lines from the standard query pattern file, `NS-query.pat` show how these work together:

```
<input type="hidden" name="NS-max-records"
value="$$NS-max-records"

<td align=left colspan=2$$logo</td
<td align=right<h3$$sitename</h3</td

<td align=right<b$$queryLabel</b</td
<td align=left&nbsp;<input name="NS-query" size=40
value="$$NS-display-query"</td
```

Each line contains standard HTML tags, and one or more variables with the `$$` or `$$NS-prefix`. Examining each line more closely requires looking at the configuration files mentioned in “Configuring Files Manually” on page 243.

- `NS-max-records`: Because this field is hidden, users cannot change this value, which defines how many matching documents to return at a time. In the advanced HTML query pattern file, `NS-advquery.pat`, this is a user-modifiable input field.
- `$$NS-max-records`: The search generates a variable from this field that can be used in subsequent searches to calculate how many result records to display at a time. In the advanced query, this value could vary for each query.
- `$$logo`: Defined in the `userdefs.ini` file. This could be any image or text the user wanted to display on the form.

- `$$sitename`: Defined in the `userdefs.ini` file as the server's host name that is provided by the `$$NS-host` search macro.
- `$$queryLabel`: Defined in the `userdefs.ini` file as a text label for the query input field. In this case, the label on the form is the word "For:"
- `NS-query`: Defined in this pattern file as the name of the input field. `$$NS-display-query`: Defined in the `userdefs.ini` file. The search generates a variable from this field that can be used in subsequent searches to determine which word or phrase to highlight when an entire matching document is displayed.

Search Function Syntax

The search function uses standard URL syntax with a series of name-value pairs for the search arguments. This is the basic syntax:

```
http://server_root/search?name=value[&name=value][&name=value]
```

As you use the HTML search query and results pages, you can see search functions and arguments displayed in the URL field of your browser. When entered directly into the URL field, these are sometimes called *decorated URLs*. They can also be embedded in your pattern files with the HREF tag.

You can create a complete search function as an HREF element within a pattern file. The example given is from the `HTML-descriptions.pat file`, which defines how collection information is displayed. The following lines produce a heading for each collection with the label ("Collection:"), and provide a link to the actual collection file through the collection's label (`NS-collection-alias`) defined in the `dblist.ini` file.

```
<td colspan=6<font size=+2<b$$collectionLabel</b
<a
href=$$NS-server-url/search?NS-collection=$$NS-collection$$NS-co
llection-alias</a
</font</td
```

The HREF contains a complete search function by using the following elements:

- `$$NS-server-url`: A search macro that determines the user's server URL. `/search`: The search command itself.
- `?`: The query string indicator. Everything after the ? is information used by the search function.
- `NS-collection=$$NS-collection`: This uses the search macro `$$NS-collection` to define the collection's filename.

You can set up a search to use a variable conditionally; if there is no value associated with the variable, nothing will be displayed. The syntax is as follows:

```
variableName[conditionalized output]
```

For example, you could request that the document's title be output if it exists. If there is no title for this document, not even the label "Title:" is to be displayed. To do this, you might enter:

```
$$Title[<PTitle: <B$$Title</B]
```

URL Encodings

When you construct HTML instructions, whether in decorated URLs or within a pattern file, you need to follow the rules for URL encoding. Any character that might be misunderstood as part of a URL should be encoded with the format of *%nn* format, where *nn* is a hexadecimal code. Blanks are converted to the + symbol (plus sign) in queries or to %20 in output. The following table shows the most commonly used URL codes.

Table 12-6 Common URL Encodings

Character	Description	Code
	Space	%20
;	Semicolon	%3B
/	Slash	%2F
?	Question mark	%3F
:	Colon	%3A
@	At sign	%40
=	Equal sign	%3D
&	Ampersand	%26

Required Search Arguments

Although you can customize almost every aspect of query and result pages, there are some arguments required for search functions to display the different types of search pages. These arguments are required whether the search function is in a decorated URL, or embedded as an HREF in a pattern file.

Search functions that display the search query page require these arguments:

- Search query (the word, phrase, or attribute you want to search on)
- Collection (can specify more than once for multiple-collection searches)

Search functions that display the search results page require these arguments:

- `NS-search-page=results` (or `r`, in upper- or lowercase)
- Collection (can be specified more than once for multiple-collection searches) search query

Search functions that display a highlighted document require these arguments:

- `NS-search-page=document` (or `d`, in upper- or lowercase)
- Document path
- Collection (can be specified only once)
- Search query (necessary if you want to highlight the query data)

Search functions that display the collection contents require only this argument:

- `NS-search-page=contents` (or `c`, in upper- or lowercase)

Using Pattern Variables

Using pattern variables you can customize the search text interface. This eliminates the need to update the actual HTML pages as user requirements change. For example, if the interface has graphics or text elements that change periodically, you can define a pattern variable pointing to a pathname where that graphic or text is maintained and stored.

There are three categories of pattern variables:

- Variables defined in the `userdefs.ini` file, to which are added a `$$` prefix in decorated URLs and pattern files. For example, `uidir`, `logo`, and `title` become `$$uidir`, `$$logo`, and `$$title`.
- Variables defined in the `dblist.ini` configuration files, having an `NS-` prefix when defined in the configuration file, and a `$$NS-` prefix when used in decorated URLs and pattern files. For example, `NS-max-records`, `NS-doc-root`, and `NS-date-time` become `$$NS-max-records`, `$$NS-doc-root`, and `$$NS-date-time`.
- Search macros and variables generated by a pattern file, which always have a `$$NS-` prefix. For example, `$$NS-host`, `$$NS-get-next`, and `$$NS-sort-by`.

User-defined Pattern Variables

You can create any number of your own user-defined pattern variables in the user definitions file, `userdefs.ini`, or you can modify existing definitions. When one of these variables is used in a pattern file, the $$ prefix is added to it. Variable names can have up to 32 characters or digits, or combinations of both. Characters can be letters A-Z in upper or lower case, hyphens (-), and underscores (_). Names are case sensitive.

The default `userdefs.ini` file included with iPlanet Web Server contains the following variables that are used to define:

- Search query pages labeled `[query]` in the file
- Results listings labeled `[toc]`
- Document display pages labeled `[record]`
- Collection contents pages labeled `[contents]`

Each line begins with a variable name, and is followed by a definition for that variable. Many are labels for screen elements, some are paths to other files, and some have more complex contents. For example, the following lines are from the query section of that file.

```
[query]
NS-character-set=iso-8859-1
uidir =  $$NS-server-url/search-ui
icondir = $$uidir/icons
l10nicondir = $$uidir/icons
htmldir = $$uidir/text
logo = <img src="$$icondir/magnifier.jpg" border=0 align=absmiddle><b><font
size=+2>N</font><font size=+1>etscape </font><font size=+2>S</font><font
size=+1>earch</font></b>
sitename = $$NS-host
help = /help/search.htm
title = Sample Search Interface
searchButtonLabel = Search
searchNote = To search, choose a collection, then enter words and phrases,
separated by commas<br>(e.g., search, jet engines, basketball).
advSearchNote = To search, choose collections, then enter words and phrases,
separated by commas<br>(e.g., search, jet engines, basketball).<p>Sorting is
done on any defined attributes. Use ’-’ to specify descending order
sort<br>(e.g., Title,-Author,+Date)
queryLabel = For:
queryLabelSJIS = $$queryLabel
queryLabelEUC = $$queryLabel
queryLabelJIS7 = $$queryLabel
collectionLabel = Search in:
booleanLabel = Boolean
sortByLabel = Sort by:
sortByLabelSJIS = $sortByLabel
sortByLabelEUC = $sortByLabel
```

```
sortByLabelJIS7 = $sortByLabel
freetextLabel = Freetext (unavailable)
maxDocumentsLabel = Documents to return:
maxDocumentsLabelSJIS = $$maxDocumentsLabel
maxDocumentsLabelEUC = $$maxDocumentsLabel
maxDocumentsLabelJIS7 = $$maxDocumentsLabel
copyright = Copyright &#169; 1997 Netscape Communications Corporation. All
Rights Reserved.
advancedButtonLabel = Advanced Button Label
helpButtonLabel = Help Button Label
```

The file also includes references to search macros, such as `$$NS-server-url`, and can refer to other user-defined variables, as in the following lines:

```
uidir = $$NS-server-url/search-ui
icondir = $$uidir/icons
```

Search macros are described further in “Macros and Generated Pattern Variables” on page 281.

You can use any supported HTML character entity in your variable definitions. You can use entity names that are defined in the *&name*; format as well as those defined with the three-digit code in the *&#nnn*; format. In the `userdefs.ini` code sample, the entity `&nbsp;` inserts a nonbreaking space, and `&#169;` inserts a copyright symbol. Some of the more commonly used entities are in the following table:

Table 12-7 Common HTML character entities

Numeric code	Entity name	Description
&#032;		Space
&#034;	&quot;	Quotation mark
&#036;	$	Dollar sign
&#058;	-	Colon
&#060;	&lt;	Less than
&#062;	&gt;	Greater than
&#153;	-	Trademark symbol
&#160;	&nbsp;	Nonbreaking space
&#169;	&copy;	Copyright symbol
&#174;	&reg;	Registered trademark

Configuration File Variables

Some variables are defined in the system configuration and in the collection configuration files. These use a prefix of `NS-` in the configuration file to differentiate them from other markup tags in an HTML page. To use these variables as arguments to the search function, you add another prefix `$$` to the variable, as in `$$NS-date-time` and `$$NS-max-records`.

Variables that define defaults for all searches on a server are defined in the system configuration files.

```
NS-max-records = 20
NS-query-pat = /text/NS-query.pat
NS-ms-tocstart = /text/HTML-tocstart.pat
NS-ms-tocend = /text/HTML-tocend.pat
NS-default-html-title = (Untitled)
NS-HTML-descriptions-pat = /text/HTML-descriptions.pat
NS-date-time = %b-%d-%y %H:%M
```

Although installations may vary depending on how each server is configured, the most commonly found variables are listed in the following table:

Table 12-8 Commonly found variables

Variable	Description
`NS-default-html-title`	The name given to HTML documents that do not contain a user-defined title. Typically set to "(Untitled)."
`NS-date-time`	The date and time format to use when displaying results.
`NS-date-input-format`	The format for inputting dates (the default is MMDDYY).
`NS-HTML-descriptions-pat`	The pattern file to use when displaying the contents of the collections.
`NS-largest-set`	The maximum number of records that can be handled as matching the search criteria. The records are displayed in groups of `NS-max-records.`
`NS-max-records`	The maximum size of the result set displayed at one time.
`NS-ms-tocend`	The pattern file to use for the footer at the bottom of the search results page when searching multiple collections.
`NS-ms-tocstart`	The pattern file to use for the header at the top of the search results page when searching multiple collections.
`NS-query-pat`	The query pattern file used when creating a query page.

Table 12-8 Commonly found variables *(Continued)*

Variable	Description
`NS-search-type`	The type of search to perform. Only Boolean is permitted.

Collection-specific variables are defined in the `dblist.ini` file. Among the variables defined there are:

```
NS-doc-root = C:/iPlanet/Servers/docs
NS-url-base = /
NS-display-select = YES
```

The variables in your `dblist.ini` file may differ according to the type of collections you are using. Table 11.9 contains some of the more commonly found collection-specific variables.

Table 12-9 Commonly found variables in dblist.ini

Variable	Description
`NS-collection-alias`	The collection's label. Can be specified more then once to search multiple collections.
`NS-doc-root`	The root directory for the documents in the collection.
`NS-display-select`	This indicates whether the collection is displayed as part of the collection information listing, when `NS-search-page=contents`. The default is YES.
`NS-highlight-start`	Begin highlighting at this point in the displayed document. Typically this highlights the search query criteria.
`NS-highlight-end`	End highlighting at this point in the displayed document.
`NS-language`	The language of the documents in the collection.
`NS-record-pat`	The pattern file to use when displaying a highlighted document page.
`NS-tocend-pat`	The footer pattern file associated with a collection to be used when formatting the search results.
`NS-tocrec-pat`	The record pattern file associated with a collection to be used when formatting the search results.

Table 12-9 Commonly found variables in dblist.ini *(Continued)*

Variable	Description
`NS-tocstart-pat`	The header pattern file associated with a collection to be used when formatting the search results.
`NS-url-base`	The base URL used when constructing the link used to locate the file.

Macros and Generated Pattern Variables

There are some search macros that you can use in your pattern files or decorated URLs. The search function itself generates some pattern variables you can use in subsequent search requests to define how output is to be displayed. These macros and variables have a prefix of `$$NS-` to indicate their use.

For example, after doing an initial search query that results in 24 documents on the results page, you can reuse the search-generated `$$NS-docs-matched,` and the `$$NS-doc-number` variables to help define a document page displaying one of the documents in detail. In this way, you can tell the user that this document is number 3 of 24 documents returned for the original search.

The search macros and the generated variables that you can use in a subsequent pattern file or decorated URL are listed the following table:

Table 12-10 Macros and generated pattern variables

Variable	Description
`$$NS-collection-list`	An HTML multiple select list of all the collections in `dblist.ini` where `NS-display-select` is set to YES.
`$$NS-collection-list-dropdown`	An HTML drop-down list version of `NS-collection-list.`
`$$NS-collections-searched`	The number of collections searched for this request.
`$$NS-display-query`	The HTML-displayable version of the query that is generated for a results page.
`$$NS-doc-href`	The HTML HREF tag for the document. This provides a URL to the original source document. For email, this is in the form `mailbox:/boxname?id-messageID` and for news, it is in the form `news:messageID`.
`$$NS-doc-name`	The document’s name.
`$$NS-doc-number`	The sequence number of the document in the results page list.
`$$NS-doc-path`	The absolute path to the document.

Table 12-10 Macros and generated pattern variables *(Continued)*

Variable	Description
`$$NS-doc-score`	The ranked score of the document (ranges 0 to 100).
`$$NS-doc-score-div10`	The ranked score of the document (ranges 0 to 10).
`$$NS-doc-score-div5`	The ranked score of the document (ranges 0 to 5).
`$$NS-doc-time`	The creation time for a document in the results list. To obtain this value, you must set `NS-use-system-stat` = `YES`. By default it is set to NO, since system statistics are expensive.
`$$NS-doc-size`	The size of the document rounded to the nearest K. To obtain this value, you must set `NS-use-system-stat` = `YES`. By default it is set to NO, since system statistics are expensive.
`$$NS-docs-found`	The actual number of documents that the search engine found for this request.
`$$NS-docs-matched`	The number of documents returned from the search (up to `NS-max-records`) for this request.
`$$NS-docs-searched`	The number of documents searched through for this request.
`$$NS-get-highlighted-doc`	This provides the URL for a highlighted document in order to be able to display the document as HTML text with highlights.
`$$NS-get-next`	This variable gets the next set of search results to be displayed. The set is equal to `NS-max-records` and is positioned by using `NS-search-offset`.
`$$NS-get-prev`	This variable gets the previous set of search results that has been displayed. The set is equal to `NS-max-records` and is positioned by using `NS-search-offset`.
`$$NS-host`	The host name.
`$$NS-insert-doc`	A placeholder used in the `NS-record-pat` pattern files for HTML to indicate where the source document is to be inserted.
`$$NS-rel-doc-name`	The relative name of the document to display creating a document page.
`$$NS-search-offset`	The offset into the set of records returned as search results. Used to determine which set of records are displayed when you use `NS-get-next` and `NS-get-prev`.
`$$NS-server-url`	The URL for the server.
`$$NS-sort-by`	The sort sequence for the items on the results page. You can select one or more of the available attributes for the collection. The default is an ascending sort.

Part 4

Managing Virtual Servers and Services

Chapter 13

Using Virtual Servers

This chapter explains how to set up and administer virtual servers using your iPlanet Web Server.

This chapter contains the following sections:

- Virtual Servers Overview
- Using iPlanet Web Server Features with Virtual Servers
- Using the Virtual Server User Interface
- Setting Up Virtual Servers
- Allowing Users to Monitor Individual Virtual Servers
- Deploying Virtual Servers

Virtual Servers Overview

When you use virtual servers you can offer companies or individuals domain names, IP addresses, and some server monitoring capabilities with a single installed server. For the users, it is almost as if they have their own web servers, though you provide the hardware and basic web server maintenance.

NOTE If you are not using virtual servers, you still use the items in the Class Manager to configure content, programs, and other features for your web server instance. When you install the web server, a default virtual server for the instance is created. You manage the content and services for this default virtual server using the virtual server user interface.

To set up virtual servers, you need to set up the following:

- Virtual Server Classes
- Listen Sockets
- Connection Groups
- Virtual Servers

The settings for virtual servers are stored in the `server.xml` file, found in the *server_root/server_ID/*`config` directory. You do not need to edit this file to use virtual servers, but you can. If you would like to learn more about this file and how to edit it, see the *NSAPI Programmer's Guide*.

This section includes the following topics:

- Multiple Server Instances
- Virtual Server Classes
- Listen Sockets
- Connection Groups
- Virtual Servers
- Virtual Server Selection for Request Processing
- Document Root
- Log Files
- Migrating Virtual Servers from a Previous Release

Multiple Server Instances

In past releases of the iPlanet Web Server, unique configuration information for virtual servers was not very flexible. Quite often users created separate server instances in order to have a straightforward way to have servers with separate configuration information. With iPlanet Web Server 6.0, each virtual server class has separate configuration information. Multiple server instances are still supported, but if your goal is to have many servers with separate configuration information, virtual servers are a better choice.

Virtual Server Classes

Virtual servers are grouped into classes. Using classes you can configure similar virtual servers at the same time, so you don’t have to configure each one separately. Though all virtual servers in a class share the same basic configuration information, you can also set variables and change configuration per virtual server. If you don’t want virtual servers to share configuration information, you can create a single virtual server per virtual server class. However, if your virtual servers share similar properties, you can group them in a class and configure them together.

For example, if you work for an Internet Service Provider (ISP) and want to provide different levels of hosting for different customers at different prices, you can set up several classes of virtual servers for your customers. You might enable Java servlets and JSPs for one class of virtual servers, and disable Java servlets and JSPs for a less expensive class of virtual servers.

You create a class of virtual servers by naming it and setting up a document root, where all virtual servers belonging to the class will have their document roots by default. You can use the `$id` variable so that each virtual server within the class will have a separate document root within the class’ document root. For more information, see “Document Root,” on page 292.

After creating the class of virtual servers, you associate services with it. You can turn on or configure the following types of services for a class of virtual servers:

- Programs, see Chapter 15, “Extending Your Server With Programs.”
- Content Management, see Chapter 16, “Content Management.”
- Configuration Styes, see Chapter 17, “Applying Configuration Styles.”

The obj.conf File

All virtual servers in a class share an `obj.conf` file, which stores information about the virtual server class. Some of that information is stored in variables, so that individual virtual servers can have specific variable values substituted on the fly.

For more information about `obj.conf` and variables, see the *NSAPI Programmer’s Guide*. For more information on using variables in the user interface, see “Using Variables,” on page 296.

Virtual Servers in a Class

A virtual server that belongs to a class is called a member of that class. Some virtual server settings are configured for all virtual servers in a class, and some are configured individually. These settings are configured on the Class Manager’s Virtual Servers tab. For more information, see Chapter 14, “Creating and Configuring Virtual Servers.”

The Default Class

When you install iPlanet Web Server, the installer automatically creates a single class, called `defaultclass`. It contains one virtual server member by default for your server instance. You can add additional virtual servers to the default class, but you cannot delete your default virtual server from the class. You also cannot delete the default class.

Listen Sockets

Connections between the server and clients happen on a listen socket. Each listen socket you create has an IP address, a port number, a server name, and a default virtual server (which becomes associated with the connection group created automatically for the listen socket). If you want a listen socket to listen on all configured IP addresses on a given port for a machine, use 0.0.0.0, any, ANY, or INADDR_ANY for the IP address.

When you install iPlanet Web Server, one listen socket, ls1, is created automatically. This listen socket uses the IP address 0.0.0.0 and the port number you specified as your HTTP server port number during installation (the default is 80). You cannot delete the default listen socket. If you are not using virtual servers, this one listen socket is sufficient. However, if you are using virtual servers, you may want to create multiple listen sockets for your virtual servers.

Since a listen socket is a combination of IP address and port number, you can have multiple listen sockets with the same IP address and different port numbers, or with different IP addresses and the same port number. For example, you could have 1.1.1.1:81 and 1.1.1.1:82. Additionally, you could have 1.1.1.1:81 and 1.2.3.4:81, as long as your machine is configured to respond to both these addresses. However, if you use the 0.0.0.0 or ANY IP address, which listens to all IP addresses on a port, you cannot set up additional IP addresses that listen on the same port for a specific IP address. For example, if you have a listen socket 0.0.0.0:80 (all IP addresses on port 80) you cannot also have 1.2.3.4:80.

In addition, you specify the number of acceptor threads (sometimes called accept threads) in the listen socket. Accept threads are threads that wait for connections. The threads accept connections and put them in a queue where they are then picked up by worker threads. Ideally, you want to have enough accept threads so that there is always one available when a new request comes in, but few enough so that they do not provide too much of a burden on the system. The default is 1. A good rule is to have one accept thread per CPU on your system. You can adjust this value if you find performance suffering.

Connection Groups

Each listen socket has at least one connection group associated with it. When you create a listen socket, a connection group is also created which contains the default virtual server you specified for the listen socket. The IP address for this connection group is `default`.

If your listen socket has an IP address of 0.0.0.0 or ANY, you can add multiple connection groups that respond to particular IP addresses. Using this functionality, you can set up virtual server with dedicated IP addresses.

For each listen socket, there is always a connection group that is the default connection (the IP address shows as `default`). If the listen socket has a specific IP address, this default connection group is the only one available. If the listen socket listens on any IP address, the default connection group is the one used if the request doesn't find a specific IP address match among the other connection groups in the listen socket.

When you install your server, one connection group is created by default for the default listen socket ls1. The IP address for the connection group is `default`, the port is 80, and the default virtual server is the default server created when you installed.

For each virtual server you select the connection group or groups it responds to. You do not do this for the class as a whole: the connection group information is independent of the virtual server classes.

Virtual Servers

To create a virtual server you must first decide which class you want it to belong to. Next you need to decide what kind of virtual server you want. To create a virtual server, all you need to specify is a virtual server ID, one or more connection groups, and one or more URL hosts.

This section includes the following topics:

- Types of Virtual Servers
- IP-Address-Based Virtual Servers
- URL-Host-Based Virtual Servers
- Default Virtual Server

Types of Virtual Servers

In previous versions of iPlanet Web Server, there were two kinds of virtual servers: hardware and software. Hardware virtual servers had unique IP addresses associated with them. Software virtual servers did not have unique IP addresses but instead had unique URL hosts.

In iPlanet Web Server 6.0, these concepts are no longer quite accurate. All virtual servers have a URL host specified. However, the virtual server may also be associated with an IP address based on its listen socket and connection group information.

When a new request comes in, the server determines which virtual server to send it to based on the IP address or the value in the Host header. It evaluates the IP address first. For more information, see "Virtual Server Selection for Request Processing," on page 291.

IP-Address-Based Virtual Servers

In order to have multiple IP addresses on a single computer, you must either map them through the operating system or provide additional cards. To set up multiple IP addresses through the operating system, use the Network Control Panel (Windows NT) or the `ifconfig` utility (Unix/Linux). Please note that directions for using `ifconfig` vary from platform to platform. Consult your operating system documentation for more information.

Typically you create an IP-address-based virtual server by creating a listen socket that listens on any IP address, and creating additional connection groups for each listen socket. Each of these connection groups has a specific IP address. You then associate a virtual server as the default virtual server for each connection group. However, you can also create an individual listen socket for each IP address. For more information on ways to deploy virtual servers, see "Deploying Virtual Servers," on page 304."

URL-Host-Based Virtual Servers

You can set up URL-host-based virtual servers by giving them unique URL hosts. The contents of the Host request header directs the server to the correct virtual server.

For example, if you want to set up virtual servers for customers *aaa*, *bbb*, and *ccc*) so that each customer can have an individual domain name, you first configure DNS to recognize that each customer's URL, `www.aaa.com, www.bbb.com`, `www.ccc.com` , resolves to the IP address of the listen socket you are using. You then set the URL hosts for each virtual server to the correct setting (for example, `www.aaa.com`).

You can have any number of these URL-host-based virtual servers associated with a connection group.

Because URL-Host-based virtual servers use the Host request header to direct the user to the correct page, not all client software works with them. Older client software that does not support the HTTP Host header won't work. These clients will receive the default virtual server for the connection group.

Default Virtual Server

URL-Host-based virtual servers are selected using the Host request header. If the end user's browser does not send the Host header, or if the server cannot find the specified Host header, the default virtual server services the request.

Also, for IP-address-based virtual servers, if iPlanet Web Server cannot find the specified IP address, the default virtual server services the request. You can configure the default virtual server to send an error message, or server pages from a special document root.

NOTE Do not confuse the default virtual server for a connection group with the default class and virtual server created when you install the server. The default class is created at installation time and contains the server instance's virtual server as a member. The default virtual server for a connection group is any virtual server you designate as the default.

The default virtual server is set by connection group. You specify a default virtual server when you create a listen socket. That becomes the default virtual server of the connection group created by default for the listen socket. You can always change the default virtual server. If the listen socket has a specific IP address associated with it, this connection group is the only connection group for the listen socket. If the listen socket listens on any IP address, each connection group you create will have a default virtual server.

Virtual Server Selection for Request Processing

Before the server can process a request, it must accept the request via a listen socket, then direct the request to the correct connection group and virtual server.

A connection group is first selected as follows:

- If the listen socket is configured to listen on a particular IP address, it can contain only one connection group, and that group is selected.
- If the listen socket is configured to listen on ANY, the IP address to which the client connected is matched to the IP address of a connection group contained by that listen socket. If no IP address matches, the default connection group with `default` as the IP address is selected.

A virtual server is then selected as follows:

- If the connection group is configured to only a default virtual server, that virtual server is selected.
- If the connection group has more than one virtual server configured to it, the request `Host` header is matched to the URL host of a virtual server. If no `Host` header is present or no URL host matches, the default virtual server for the connection group is selected.

If a virtual server is configured to an SSL listen socket, its URL host is checked against the subject pattern of the certificate at server startup, and a warning is generated and written to the error log if they don't match.

After the virtual server is determined, the server executes the `obj.conf` file for the virtual server class to which the virtual server belongs. For details about how the server decides which directives to execute in `obj.conf`, see the *NSAPI Programmer's Guide*.

Document Root

The primary document directory or document root is the central directory that contains all the virtual server's files to make available to remote clients.

The document root directory provides an easy way to restrict access to the files on a virtual server. It also makes it easy to move documents to a new directory (perhaps on a different disk) without changing any of the URLs because the paths specified in the URLs are relative to the primary document directory.

For example, if your document directory is `C:\iplanet\servers\docs`, a request such as `http://www.iplanet.com/products/info.html` tells the server to look for the file in `C:\iplanet\servers\docs\products\info.html`. If you change the document root (that is, you move all the files and subdirectories), you only have to change the document root that the virtual server uses, instead of mapping all URLs to the new directory or somehow telling clients to look in the new directory.

When you install the iPlanet Web Server, you designate a document root for your web server instance. That becomes the document root for the default class. You can change that directory at the class level or override it at the individual virtual server level.

When you add a class, you also need to specify a document directory. That directory is an absolute path. However, if you simply enter an absolute path, the document roots for all virtual servers belonging to the class default to the same directory. If you include the variable $id at the end of your document root absolute path, every virtual server has a default document root of *class_doc_root*/*virtual_server_ID*. For example, if your class' document directory is `/iplanet/servers/docs/$id`, the default document directory for a virtual server `vs1` that belongs to the class is `/iplanet/servers/docs/vs1`.

For more information on variables, see "Using Variables," on page 296."

You can also override the class' default document directory at the individual virtual server level.

Log Files

When you create a new virtual server, by default the log file is the same log file as the server instance. In most cases you will want each individual virtual server to have its own log file. To set this up, you can change the log path for each virtual server.

For more information, see "Configuring Virtual Server Log Settings," on page 320.

Migrating Virtual Servers from a Previous Release

If you used virtual servers in the 4.x version of iPlanet Web Server, you can migrate them to the current release using the migration tools. For more information, see the *Installation and Migration Guide*.

Using iPlanet Web Server Features with Virtual Servers

iPlanet Web Server has many features, such as SSL and access control, that you can use with virtual servers. Many of these features involve configuration for all servers, for a server instance, for a class of virtual servers, or an individual virtual server. The following sections describe the features and provide information on where to look for more information.

This section includes the following topics:

- Using SSL with Virtual Servers
- Using Access Control with Virtual Servers
- Using CGIs with Virtual Servers
- Using Configuration Styles with Virtual Servers

Using SSL with Virtual Servers

If you want to use SSL on a virtual server, in most cases you use an IP-address-based virtual server. The customary port is 443. It is difficult to use SSL on a URL-host-based virtual server because iPlanet Web Server must read the request before determining which URL host to send the request to. Once the server reads the request, the initial handshake, where security information is exchanged, has already happened.

The only exception is when URL-Host-based virtual servers all have the same SSL configuration, including the same server certificate, using “wildcard certificates.” For more information, see Chapter 5, “Securing Your Web Server.”

One way to implement SSL with virtual servers is to have two listen sockets, one using SSL and listening to port 443, and one that is not using SSL. A user would typically access the virtual server through the non-SSL listen socket. When the need to have secure transactions arises, users could click a button on the web page to start initiating secure transactions. After that, the requests go through the secure listen socket.

Because SSL transactions are much slower than non-SSL transactions, this design limits the SSL transactions to only the ones that are necessary. Faster, non-SSL connections are used the rest of the time.

For more information on setting up and using security with you iPlanet Web Server and virtual servers, see Chapter 5, “Securing Your Web Server.” For a diagram of a sample SSL configuration with virtual servers, see “Example 2: Secure Server,” on page 306.

Using Access Control with Virtual Servers

With virtual servers you have the ability to set up access control on a per virtual server basis. You can even configure it so that each virtual server can have user and group authentication using an LDAP database. For more information, see “Controlling Access for Virtual Servers,” on page 193.

Using CGIs with Virtual Servers

You can use CGIs on virtual servers. There are many settings that you can configure on for access and security reasons.

For more information on setting up and using CGIs, see “Installing CGI Programs,” on page 333.

Using Configuration Styles with Virtual Servers

Configuration styles are an easy way to apply a set of options to specific files or directories that your various virtual servers maintain. For more information on using configuration styles see Chapter 17, “Applying Configuration Styles.”

Using the Virtual Server User Interface

To create and edit virtual servers, you can use the user interface or a command line utility.

The user interface for administering virtual servers has three parts:

- The Server Manager contains settings that affect the server as a whole (or all virtual servers).
- The Class Manager contains settings that affect a single class and the virtual servers within the class.
- The Virtual Server Manager contains settings for an individual virtual server.

In addition, a user interface for end-users who have an individual virtual server is available. For more information, see “Allowing Users to Monitor Individual Virtual Servers,” on page 301.

This section includes the following topics:

- The Class Manager
- The Virtual Server Manager
- Using Variables
- Dynamic Reconfiguration

The Class Manager

To access the Class Manager follow these steps:

1. From the Server Manager, click the Virtual Server Class Tab.
2. Click Manage Classes.
3. Choose a class and click Manage.

You can also click the class name in the tree view of the server, or click the Class Manager button link in the upper right corner of the Server Manager.

The Virtual Server Manager

To access the Virtual Server Manager, follow these steps:

1. From the Class Manager, click the Virtual Server Tab.
2. Click Manage Virtual Servers.
3. Choose a virtual server and click Manage.

You can also click the virtual server name in the tree view of the server.

You can use a command line utility, `HttpServerAdmin`, to perform the same virtual server tasks as you can perform using the user interface. For more information on the command line utility `HttpServerAdmin`, see "HttpServerAdmin (Virtual Server Administration)," on page 370.

Using Variables

You can use variables to give virtual-server specific values for a class without having to define each value individually. A variable is defined in the `obj.conf` file. You can define your own variables, but the user interface will not recognize them. The variable that is most useful in the user interface is the variable $id, which represents the ID of the virtual server. Whenever you enter this variable, the server substitutes the value for the individual virtual server ID.

There are a few other variables, such as $accesslog (the path to each virtual server's access log) and $docroot (the path to each virtual server's document root), that you may occasionally see, but $id is the only one you should need to enter into a field.

For more information on variables, see the *NSAPI Programmer's Guide*.

Dynamic Reconfiguration

Dynamic reconfiguration allows you to make configuration changes to a live web server without having to stop and restart the web server for the changes to take effect. You can dynamically change all configuration settings and attributes in `server.xml` and its associated files without restarting the server. So any changes that you make within the virtual server user interface can by applied without restarting the server. You can dynamically reconfigure your server after changes using the reconfiguration script or the user interface.

On Unix platforms the dynamic reconfiguration script is a shell script named ’`reconfig`’ located in each instance’s directory. There are no commandline arguments to this script. You can run the reconfiguration script by simply typing ’`reconfig`’ from the server instance’s directory.

On NT, the dynamic reconfiguration script is a batch file called ’`reconfig.bat` located in each instance’s directory. There are no command line arguments. You can run the reconfiguration script by simply typing ’`reconfig`’ or ’`reconfig.bat`’ from the server instance’s directory.

When run, this script initiates a dynamic reconfiguration of the server, similar to the user interface, and displays the server messages related to reconfiguration.

To access the dynamic reconfiguration screen, click the Apply link found in the upper right corner of the Server Manager, Class Manager, and Virtual Server Manager pages, then click the Load Configuration Files button on the Apply Changes page. If there are errors in installing the new configuration, the previous configuration is restored.

Setting Up Virtual Servers

To set up virtual servers, follow these steps:

1. Create a listen socket
2. Create a connection group
3. Create a class of virtual servers
4. Configure the services for the class
5. Create the virtual servers in a virtual server class
6. Configure virtual servers

Please note that you must enter an existing virtual server in the default virtual server field when you create a listen socket. You can use the virtual server created when you installed the server, and then go back and change it after you've created additional virtual servers, if you like.

Creating a Listen Socket

To create a listen socket, follow these steps:

1. From the Server Manager, click the Preferences tab.
2. Click Add Listen Socket.
3. Fill in the fields.

   Listen sockets must have a unique combination of port number and IP address. You can use either IPV4 or IPV6 addresses. If you want to create a listen socket for IP-address-based virtual servers, the IP address must be 0.0.0.0, ANY, any or INADDR_ANY, meaning it listens on all IP addresses on that port. You can then specify a particular IP address at the connection group.

   You can also enable security (SSL) for this listen socket.

   The Server Name field specifies the host name in the URLs the server sends to the client. This affects URLs the server automatically generates; it doesn't affect the URLs for directories and files stored in the server. This name should be the alias name if your server uses an alias.

   The default virtual server is the virtual server that will answer requests for the listen socket's default connection group if no other virtual server is found first.

   For more information, see "Virtual Server Selection for Request Processing," on page 291.
4. Click OK.

Creating a Connection Group

When you add a listen socket, a default connection group is added automatically. If your listen socket responds to any IP address, you can add additional connection groups.

To add a connection group, follow these steps:

1. From the Server Manager, click the Preferences tab.
2. Click Edit Listen Sockets.

3. On the line next to the listen socket to which you want to add a connection group, click Groups.

   A list of the all groups associated with the listen socket appears.

4. To add a group, use the line at the top of the screen where the action is set to Add and fill in the fields.

   If your listen socket has a single IP address, you won’t be able to add an additional connection group.

Creating a Virtual Server Class

To create a virtual server class, follow these steps:

1. From the Server Manager, click the Virtual Server Class tab.

2. Click Add Class.

3. Name the class.

4. Insert a document root for the class.

   The directory must already exist. All virtual servers for this class will have document roots in this absolute path, unless you specify otherwise. If you use `/$id` as the last part of the path, a document root folder named for the virtual server ID is automatically created within the class’ document root path.

5. Click OK.

Once you have created a class of virtual servers, choose the services associated with the class. For more information, see Chapter 16, “Content Management.”

Editing or Deleting a Virtual Server Class

To edit a virtual server class’s settings, follow these steps:

1. From the Server Manager, click the Virtual Server Class tab.

2. Click Edit Classes.

3. From the pull-down list next to the class you want, choose Edit or Delete.

   Please note that you cannot delete the default class.

4. Use the Document Root field to change to absolute path to the class' default document root.

   The document roots for virtual servers in this class are created within this directory by default.

5. Enter On in the Accept Language field if you wish this class of virtual servers to use accept language header parsing.

   The default is Off.

6. If you want to change the CGI defaults associated with a class, click Advanced.

   A window with the CGI defaults appears. Edit the fields and click OK to return to the Edit a Class window. The Reset button rolls back your changes.

7. Click OK. The class is changed or deleted.

Specifying Services Associated with a Virtual Server Class

Some of the characteristics that differentiate one class of virtual servers from another are the services that are enabled for that class of virtual servers. For example, one class of virtual servers might have CGIs enabled while another doesn't. For more information on setting up services, see Chapter 16, "Content Management."

Creating a Virtual Server

Once you have set up a virtual server class, you can create a virtual server. Because virtual servers are members of a particular virtual server class, you create virtual servers on the Class Manager.

For more information, see "Creating a Virtual Server," on page 313.

Specifying Settings Associated with a Virtual Server

You can override some class settings at the virtual server level and also configure additional settings. You configure these settings in the Class Manager.

For more information, see Chapter 14, "Creating and Configuring Virtual Servers.

Allowing Users to Monitor Individual Virtual Servers

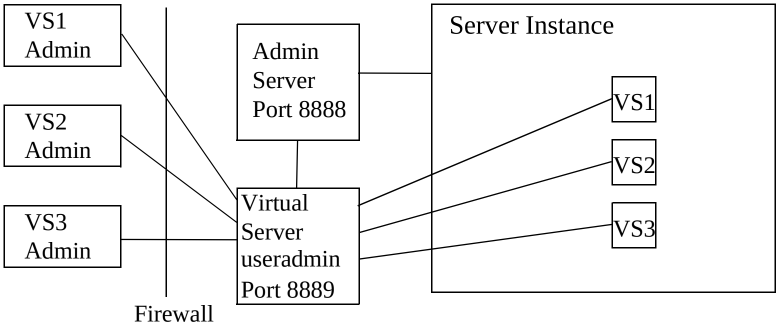
A special user interface exists for the administrators of individual virtual servers that allows them to see settings for their virtual servers and to view their access and error logs. For example, if you have an intranet with three different virtual servers for three different departments, each department can view their settings and log files individually.

For security reasons, this administration user interface is on a separate port from either the administration server port or the web server instance port.

This user interface runs on a virtual server within the administration server. This virtual server is set up by default and is called useradmin. You must set up a listen socket in the administration server that is separate from the listen socket the administration server runs on, so that people can access the virtual server administration user interface without having access to your administration server port.

Figure 13-1 shows the administrators of individual virtual servers accessing the useradmin virtual server in order to access the information for their virtual servers.

Figure 13-1 Configuring virtual server administrator's user interface



When you turn on a virtual server, users can administer it through the following URL:

server_name:*port*/user-app/*server_instance*/*virtual_server_ID*

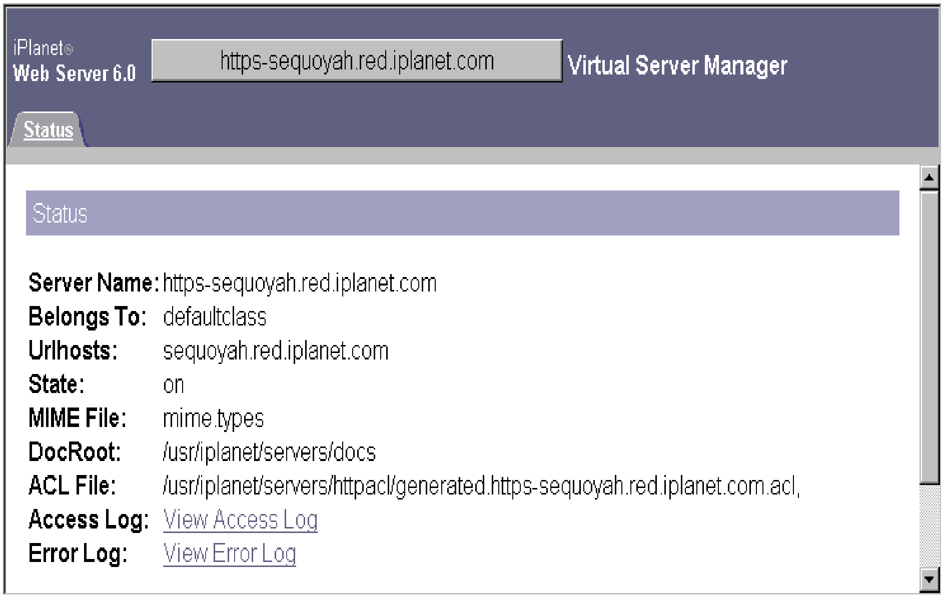
For example:

iplanet:8889/user-app/iplanet/vs2

The server instance doesn't include the "https" portion of the server instance name.

Figure 13-2 shows the user interface that the end users see:

Figure 13-2 Virtual Server Administration user interface



Status

Server Name: https-sequoyah.red.iplanet.com
Belongs To: defaultclass
Urlhosts: sequoyah.red.iplanet.com
State: on
MIME File: mime.types
DocRoot: /usr/iplanet/servers/docs
ACL File: /usr/iplanet/servers/httpacl/generated.https-sequoyah.red.iplanet.com.acl,
Access Log: View Access Log
Error Log: View Error Log

To configure your server to use this feature, follow these steps:

1. Create a new listen socket that runs a port separate from the port that the administration server uses.

   For example, if your administration server runs on port 8888, this new listen socket must have a different port number. Using a different listen socket helps safeguard your administration server.

   For security reasons, you cannot add this listen socket through the user interface. Instead, you add it in the administration server's `server.xml` file.

2. Open the administration server's `server.xml` file, found at server_root/https-admserv/config/server.xml.

3. Add a new listen socket and connection group to the file.

   The IP address should be 0.0.0.0 or ANY, and the port number should be a port different from the administration server's port. The default virtual server should be useradmin.

Code Example 13-1 New listen socket

```
<LS id="ls2" ip="0.0.0.0" port="8889" security="off" acceptorthreads="1"
blocking="no">
<CONNECTIONGROUP id="group2" matchingip="default" servername="iplanet.com"
defaultvs="useradmin"/>
</LS>
```

In this example, `ls2` is the listen socket created with a connection group of `group2`.

4. Edit the `server.xml` file so that the useradmin virtual server (found in the class userclass) uses the connection group you created.

5. Set the state for the useradmin virtual server to "on."

Code Example 13-2 Updated useradmin

```
<VSCLASS id="userclass" objectfile="userclass.obj.conf" rootobject="default" >
<VS id="useradmin" connections="group2" state="on" mime="mime1"
urlhosts="user-app" aclids="acl1">
<VARS webapps_file="user-apps.xml" webapps_enable="on"/>
<USERDB id="default" database="default" />
</VS>
</VSCLASS
```

In this example, the connection group is set to `group2`, the group created previously, and the state is set to `on`.

6. Save your changes to `server.xml`.

7. Apply the changes by restarting the Administration Server.

8. For any virtual server in any server instance, you should now be able to access the administrator UI by using the following URL:

   server_name:*port*/user-app/server_instance/virtual_server_ID

Access Control

To protect the virtual server administration from unauthorized users, you can set up ACLs. Because the URI for each virtual server is unique, you can set access so that only the correct administrator can access the settings for a virtual server.

For more information, see Chapter 8, "Controlling Access to Your Server."

Log Files

Each virtual server can have its own log files. By default, all virtual servers share the log file of the server instance. If you allow users to view their log files, in most cases you should change the log file settings so that each virtual server has its own access and error log.

For more information, see "Configuring Virtual Server Log Settings," on page 320.

Deploying Virtual Servers

iPlanet Web Server's virtual server architecture is very flexible. A server instance can have any number of listen sockets, both secure and non-secure. You can associate any number of virtual servers with these sockets through connection groups. You can have both IP-address-based and URL-host-based virtual servers.

In addition, you can group virtual servers with similar settings into any number of virtual server classes. All virtual servers in a virtual server class share the same request processing instructions in `obj.conf`.

Every virtual server can (but does not have to) have its own list of ACLs, its own `mime.types` file, and its own set of Java Web Applications.

This design gives you maximum flexibility to configure the server for a variety of applications. The following examples discuss some of the possible configurations available for iPlanet Web Server.

Example 1: Default Configuration

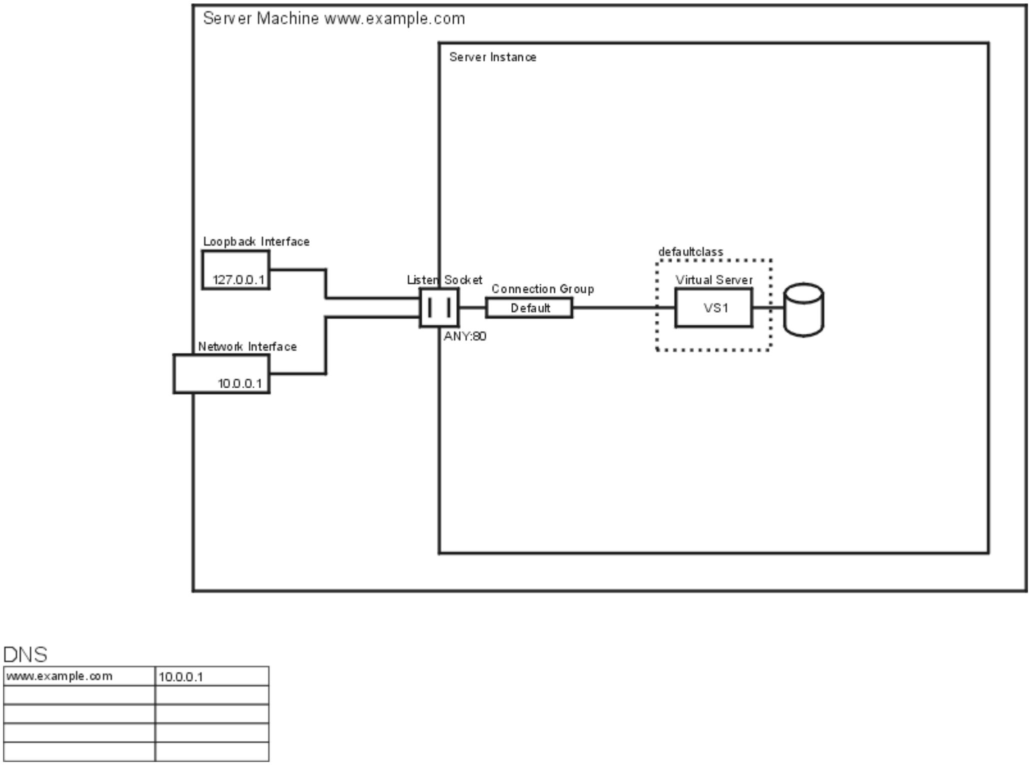
After a new installation of the iPlanet Web Server, you have one server instance. This server instance has just one listen socket listening on port 80 (or whatever you selected at installation) of any IP address to which your computer is configured.

Some mechanism in your local network establishes a name-to-address mapping for each of the addresses to which your computer is configured. In the following example, the computer has two network interfaces: the loopback interface (the interface that exists even without a network card) on address 127.0.0.1, and an ethernet interface on address 10.0.0.1.

The name example.com is mapped to 10.0.0.1 via DNS. The listen socket is configured to listen on port 80 on any address to which that machine is configured ("ANY:80" or "0.0.0.0:80").

As there are no IP-address-based virtual servers in the default configuration, the only connection group is the default one. All connections pass through to virtual server VS1.

Figure 13-3 Default configuration



www.example.com	10.0.0.1

In this configuration, connections to the following reach the server and are served by virtual server VS1

- http://127.0.0.1/ (initiated on example.com)
- http://localhost/ (initiated on example.com)
- http://example.com/
- http://10.0.0.1/

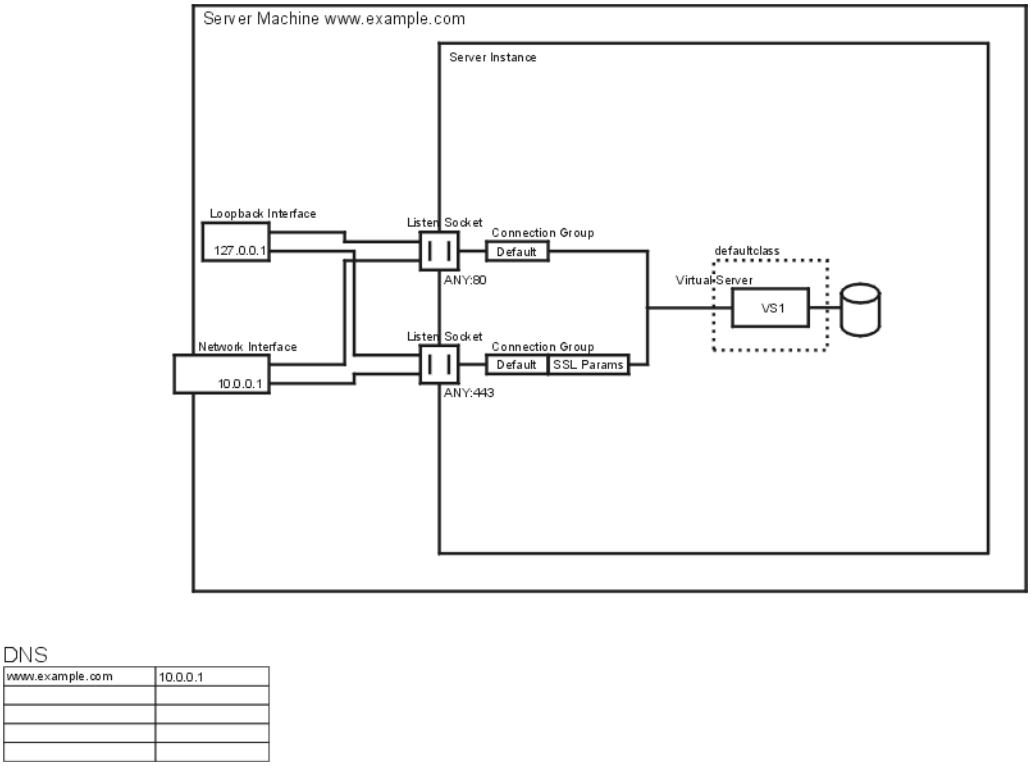
Use this configuration for traditional web server use. You do not need to add additional virtual servers or listen sockets. You configure the settings of the server by changing the settings for defaultclass (VS1 is a member of defaultclass), and VS1 itself.

Example 2: Secure Server

If you want to use SSL in the default configuration, you can simply change the listen socket to secure mode. This is a similar to the way you set security in previous versions of the iPlanet Web Server.

You can also add a new secure listen socket configured to ANY:443 and associate VS1 to the new listen socket's default connection group. The virtual server now has two connection groups, one that uses the secure listen socket, and one that doesn't. Now your server will serve the same content both with and without SSL, i.e. http://example.com/ and https://example.com/ deliver the same content.

Figure 13-4 Secure server



DNS

www.example.com	10.0.0.1

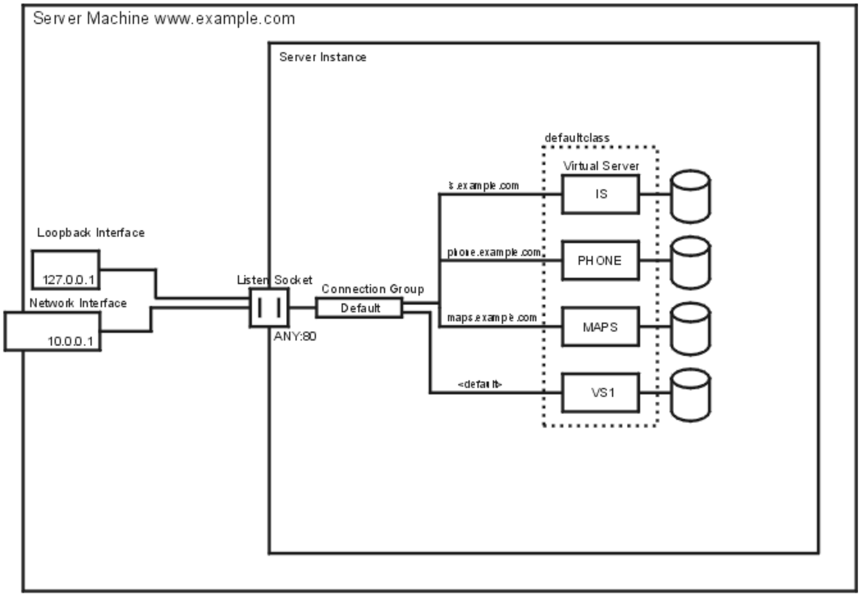
Please note that the SSL parameters are attached to the connection group. Therefore, there can only be one set of SSL parameters for all the virtual servers in a particular connection group.

Example 3: Intranet Hosting

A more complex configuration of the iPlanet Web Server is one in which the server hosts a few virtual servers for an intranet deployment. For example, you have three internal sites where employees can look up other users' phone numbers, look at maps of the campus, and track the status of their requests to the Information Services department. Previously (in this example), these sites were hosted on three different computers that had the names phone.example.com, maps.example.com and is.example.com mapped to them.

To minimize hardware and administrative overhead, you want to consolidate all three sites into one web server living on the machine example.com. You could set this up in two ways: using URL-host-based or IP-address-based virtual servers. Both have distinct advantages and disadvantages.

Figure 13-5 Intranet hosting using URL-host-based virtual servers



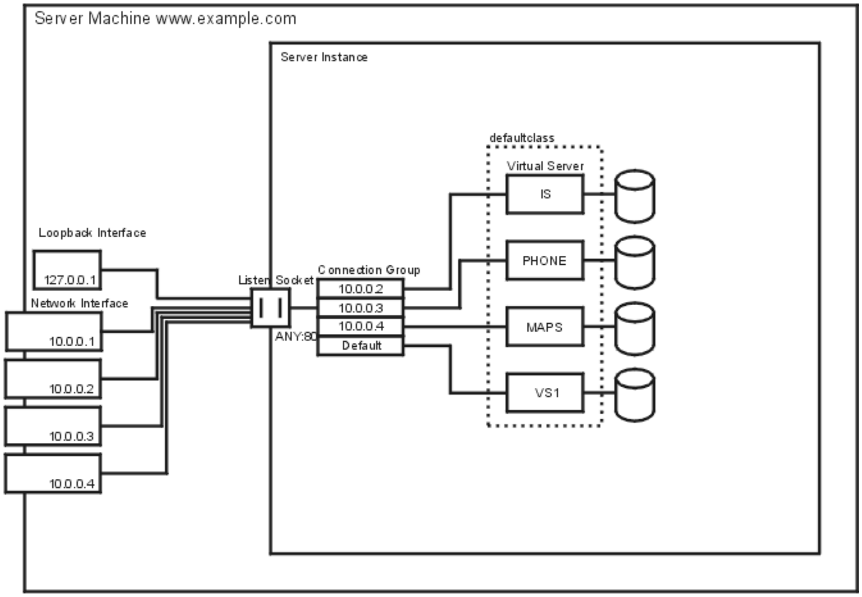
DNS

www.example.com	10.0.0.1
is.example.com	10.0.0.1
phone.example.com	10.0.0.1
maps.example.com	10.0.0.1

While URL-host-based virtual servers are easy to set up, they have the following disadvantages:

- Supporting SSL in this configuration requires non-standard setup using wildcard certificates. For more information see Chapter 5, “Securing Your Web Server.”
- URL-host-based virtual servers don’t work with legacy HTTP clients

Figure 13-6 Intranet hosting using IP-addressed-based virtual servers



DNS

www.example.com	10.0.0.1
is.example.com	10.0.0.2
phone.example.com	10.0.0.3
maps.example.com	10.0.0.4

The advantages to IP-address-based virtual servers are:

- They work with older clients that do not support the HTTP/1.1 Host header.
- Providing SSL support is straightforward.

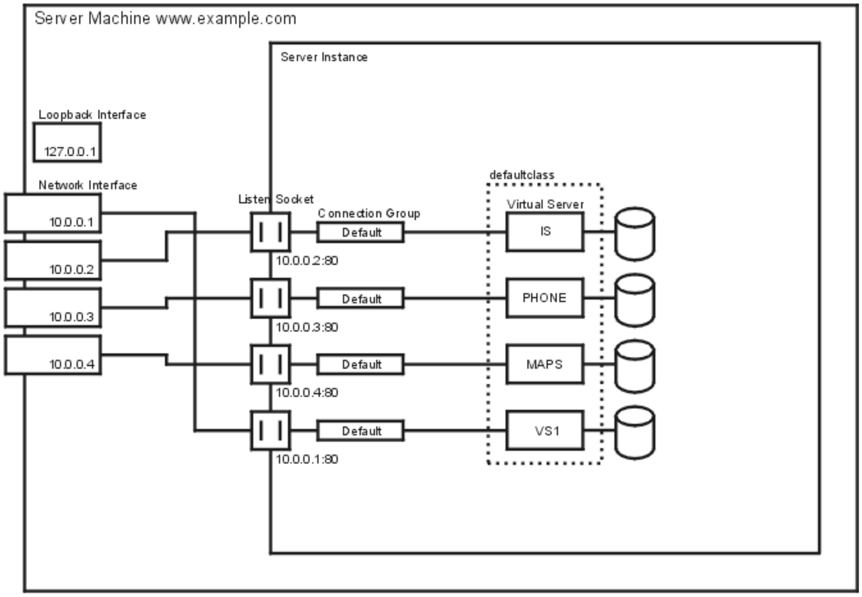
The disadvantages are:

- They require configuration changes on the host computer (configuration of real or virtual network interfaces)
- They don’t scale to configurations with thousands of virtual servers

Both configurations require setting up name-to-address mappings for the three names. In the IP-address-based configuration, each name maps to a different address. The host machine must be set up to receive connections on all these addresses. In the URL-host-based configuration, all names can map to the same address, the one the machine had originally.

As a footnote, it is also possible to set up the IP-address-based configuration with one listen socket per address:

Figure 13-7 Intranet hosting using separate listen sockets



DNS

www.example.com	10.0.0.1
is.example.com	10.0.0.2
phone.example.com	10.0.0.3
maps.example.com	10.0.0.4

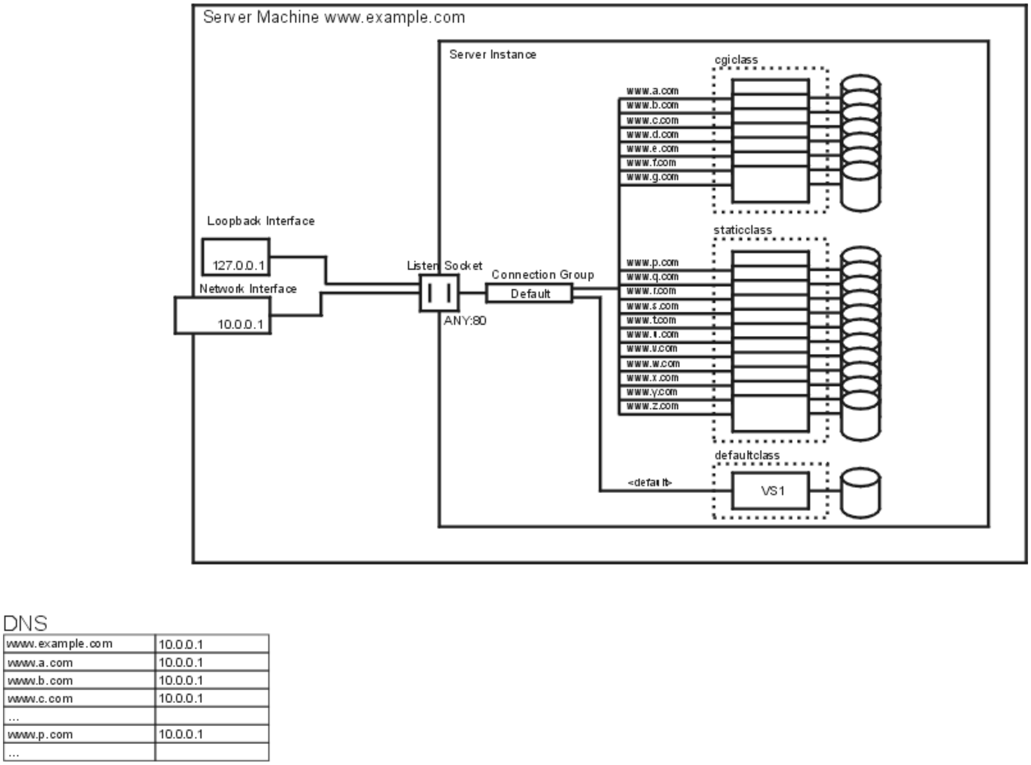
Compared to the original configuration for IP-address-based virtual servers with one listen socket on ANY:80, the configuration with multiple listen sockets may give you a minimal performance gain because the server does not have to find out the address the request came in on. However, using multiple listen sockets also results in additional overhead (memory and scheduling) because of the additional acceptor threads.

Example 4: Mass Hosting

Mass hosting is a configuration in which you enable many low-traffic virtual servers. For example, an ISP that hosts many low-traffic personal home pages would fall into this category.

The virtual servers are usually URL-host-based and are in one of multiple virtual server classes, depending on the level of service provided. For example, you could have one class that allows only static content, and another one that allows static content plus CGIs.

Figure 13-8 Mass Hosting



www.example.com	10.0.0.1
www.a.com	10.0.0.1
www.b.com	10.0.0.1
www.c.com	10.0.0.1
...	
www.p.com	10.0.0.1
...	

Notice that the virtual server installed when you installed the server, VS1, still exists in `defaultclass`.

Chapter 14

Creating and Configuring Virtual Servers

A class of virtual servers has virtual servers (members of the class) associated with it. You can override some of the class-level settings at the virtual server level. This chapter describes how you can create and configure individual virtual servers. For information on configuring virtual server classes, see Chapter 16, “Content Management.” For an overview of virtual servers, see Chapter 13, “Using Virtual Servers.”

This chapter contains the following sections:

- Creating a Virtual Server
- Editing Virtual Server Settings
- Editing Using the Virtual Server Manager
- Editing Using the Class Manager
- Deleting a Virtual Server

Creating a Virtual Server

Virtual servers allow you, with a single installed server, to offer companies or individuals domain names, IP addresses, and some server administration capabilities. For an introduction to virtual servers and how to set them up in the iPlanet Web Server, see Chapter 13, “Using Virtual Servers.”

To create a virtual server, follow these steps:

1. From the Class Manager, choose the Virtual Servers tab.
2. Click Add a Virtual Server.
3. Choose a name for the virtual server.

4. Choose a connection group for the virtual server.
5. Choose a URL host for the virtual server.

   You can type more than one URL host, separated by spaces.
6. Click OK.

These settings are all that is required for creating a virtual server. However, you can configure additional virtual server settings using other pages on this tab.

Editing Virtual Server Settings

Once you have set up your virtual servers, you can edit them. You can make these changes two ways: using the Class Manager or the Virtual Server Manager.

On the Class Manager, the pages are organized by the kind of setting you want to change. For example, you can go to the Quality of Service page to change the Quality of Service settings for one or more virtual servers in the class.

On the Virtual Server Manager, the pages only pertain to one virtual server, so you can see and change all of its settings.

Editing Using the Virtual Server Manager

The Virtual Server Manager contains three tabs: Preferences, Logs, and Web Applications.

The Preferences tab contains pages for:

- Status
- Settings

The Status page lists some settings and provides links to the virtual server's access and error logs.

The Settings page contains the following settings for a virtual server:

- State (on or off)
- Document root
- Access and error log directories
- ACL file

- MIME types file
- CGI settings

If you are editing a single virtual server, it’s convenient to use the Virtual Server Manager and change all these settings on one page.

The Logs tab contains a single page allowing you to generate reports for the selected virtual server.

The Web Applications tab contains the following pages:

- Java Web Applications Settings
- Deploy Web Application
- Edit Web Applications

The Java Web Applications Settings page displays which file the web applications are set to. `web-apps.xml` is set as the default file. This page also allows you to set the web applications state to on or off. On is set as default. This setting must be on before you can deploy or edit a web application.

The Deploy Web Application page allows you to deploy a WAR file on either your local machine or a remote server, specifying the path, URI, and directory.

The Edit Web Applications page allows you to edit, delete, or disable WAR files on a selected virtual server.

For more information about deploying and editing web application files, seeChapter 15, “Extending Your Server With Programs”. Deploying and Editing Web Applications with the User Interface.

Generating Reports for a Virtual Server

You can now generate a report for a single virtual server using the Virtual Server Manager. To do so, you should first create a new access log to be used by the virtual server, and add the new access log to the virtual server settings, as decribed below.

To generate a report for a virtual server, follow these steps:

1. Go to the Preferences tab of the Server Manager for the server instance and select Magnus Editor.
2. Select Logging Settings from the drop-down list and click Manage.

3. Set the value of `LogVSid` to On using the drop-down list.

   You can also manually set `LogVSid` to On by adding `LogVSid on` in the `magnus.conf` file.

4. Click OK.
5. Click Apply.
6. Click Apply Changes for your changes to take effect.
7. Go to the Logs tab in the Server Manager for the server instance and select Log Preferences.
8. Create a new access log by entering the path and file name in the Log File field.

   You can also manually create a new access log in magnus.conf by changing

   `Init fn=init access="$accesslog"` to `Init fn=init access="newaccesslog"`

9. Select Only Log under Format and check VSid.

   For a custom format, select Custom Format and add `%vsid%` to the end of the line.

   `%vsid%` is useful when using multiple virtual servers. This entry records vsid in the access log.

   You can also manually add `%vsid%` to the end of `Init fn` in the `magnus.conf` file.

10. Click OK.
11. Click Apply.
12. Click Apply Changes for your changes to take effect.
13. Select the virtual server you wish to generate a report for and go to the Virtual Server Manager .
14. Go to the Preferences tab and select Settings.

    In the Access Log field, change the access log to the newly created one.

15. Click OK.
16. Click Apply.
17. Click Apply Changes for your changes to take effect.

18. Select the Logs tab.

    The Generate Reports page appears.

    This page will not appear unless a virtual server has been created and `LogVSid` is On, as described above.

19. (Optional) change the settings if desired.

20. Click OK to generate the report.

Editing Using the Class Manager

Use the following Class Manager pages to edit virtual server settings.

Editing Virtual Server Settings

To edit the general settings of a virtual server, use the Edit Virtual Servers page. To access this page, follow these steps:

1. From the Class Manager, click the Virtual Servers tab.

2. Click Edit Virtual Servers.

3. From the pull-down list next to the virtual server you want, choose Edit or Delete.

   The default virtual server can only be edited and not deleted.

4. Set the State to On, Off, or Disable.

   If you set the state to Disable, you can turn the server back on, but the end user of the server cannot.

   This state is the virtual server’s state, which is independent of whether the server instance is on or off. If a virtual server’s state displayed on this page is on, the virtual server can only accept requests if the server instance is on as well.

   This is true of the default virtual server for the default server instance as well. If you turn off your server instance, your default virtual server is still set to on, but will not accept connections.

   You cannot turn off or disable the default virtual server for the server instance.

5. Select a connection group from the list in the Connections column.

6. Type the URL Hosts you want to use, if different than displayed under Urlhosts column.

   You can type more than one URL host, separated by spaces.

7. When you are through editing virtual servers click OK.

Configuring Virtual Server MIME Settings

You can set the MIME types file for an individual virtual server. The MIME types file contains the mappings of file extensions to types of files. For example, the MIME types file is where you can specify that all files ending .cgi be treated as CGI files.

You don't need to create a separate MIME types file for each virtual server or virtual server class. Instead, you create as many MIME types files as you need and associate them with a virtual server. One MIME types file, `mime.types`, exists by default on the server. To create new MIME types files, or to edit the definitions in a MIME Types file, see "Choosing MIME Types," on page 148.

To set the MIME types file for a specific virtual server, follow these steps:

1. From the Class Manager, click the Virtual Servers tab.
2. Click MIME Settings.
3. Choose a MIME types file from the pull-down menu next to the virtual server.
4. Click OK.

Configuring Virtual Server ACL Settings

You can use ACLs to control access to virtual servers. Each virtual server can have a different base DN in the LDAP database, so that each virtual server can have its own entries in a the single LDAP database used by the iPlanet Web Server.

For more information, see "Controlling Access for Virtual Servers," on page 193.

Configuring Virtual Server Security

You can set security for a virtual server if that virtual server's connection group has a secure listen socket.

For more information on security, see Chapter 5, "Securing Your Web Server."

Configuring Virtual Server Quality of Service Settings

Quality of service refers to the performance limits you set for a virtual server. For example, an ISP might want to charge different amounts of money for virtual servers depending on how much bandwidth allowed them.

You can enable these settings for the entire server or for a class of virtual servers in the Server Manager, from the Status tab. However, you can override these server or class-level settings for an individual virtual server.

Before enabling quality of service for a virtual server, you must first enable it for the entire server, and also set some basic values. See “Using Quality of Service,” on page 209.

To configure the quality of service settings for a virtual server, follow these steps:

1. From the Class Manager, click the Virtual Servers tab.
2. Click Quality of Service.

   A page appears listing all the virtual servers in the class and their quality of service settings.
3. To enable quality of service for a virtual server, choose Enable from the pull-down list.

   By default quality of service is disabled. Enabling quality of service increases server overhead slightly.
4. Set the maximum bandwidth, in bytes per second, for the virtual server.
5. Choose whether or not to enforce the maximum bandwidth setting.

   If you choose to enforce the maximum bandwidth, once the server reaches its bandwidth limit additional connections are refused.

   If you do not enforce the maximum bandwidth, when the maximum is exceeded the server logs a message to the error log.
6. Choose the maximum number of connections allowed for the virtual server.

   This number is the number of concurrent requests processed.

7. Choose whether or not to enforce the maximum connections setting.

   If you choose to enforce the maximum connections, once the server reaches its limit additional connections are refused.

   If you do not enforce the maximum connections, when the maximum is exceeded the server logs a message to the error log.

8. Click OK.

For more information on the limitations to the quality of service features, see “Using Quality of Service,” on page 209.

Configuring Virtual Server Log Settings

To change the location of the virtual server’s access and error logs from the default, follow these steps:

1. From the Class Manager, click the Virtual Servers tab.

2. Click Logging Settings.

   A page appears listing all the virtual servers in the class and the location of their error logs.

3. Enter an absolute path to the error and access logs. The path must already exist.

   By default, the access and error messages for all virtual servers are logged to the server instance’s access and error logs. If you want virtual servers to have separate log files, you set that up here.

4. If you want to change the paths back to the default, click Default.

5. Click OK.

To look at the logs for a particular virtual server, follow these steps:

1. From the Virtual Server Manager choose the Status tab.
2. Click the link to the access log or error log.
3. Choose the number of entries to display and the criteria for displaying them.

   For example, if your logs contain entries for all virtual servers, you can choose to display only the entries for a particular virtual server.
4. Click OK.

Configuring Virtual Server Java Web Application Settings

A web application is a collection of Java servlets, JSPs, HTML pages, classes and other resources. All the resources are stored in a directory, and all requests to that directory run the application. Use the Java Web Applications Settings page in the Class Manager to set web applications across all virtual servers in a class. Use the pages under the Web Applications tab of the Virtual Server Manager to deploy and edit web applications for a specific virtual server.

For more information on web applications and the `web-apps.xml` file, see Chapter 15, “Extending Your Server With Programs.”

Deleting a Virtual Server

To delete a virtual server, follow these steps:

1. From the Class Manager, click the Virtual Servers tab.
2. Click Edit Virtual Servers.
3. From the pull-down list next to the virtual server you want, choose Delete.

   You cannot delete the default virtual server that was created when you installed the server. You also cannot delete a server that is the default virtual server on a connection group.
4. Click OK. The virtual server is deleted.

Chapter 15

Extending Your Server With Programs

This chapter discusses how to install programs on the iPlanet Web Server that dynamically generate HTML pages in response to requests from clients. These programs are known as *server-side applications*. (*Client-side applications*, which are downloaded to the client, run on the client machine.)

This chapter includes the following sections:

- Overview of Server-Side Programs
- Java Servlets and JavaServer Pages (JSP)
- Installing CGI Programs
- Installing Windows NT CGI Programs
- Installing Shell CGI Programs for Windows NT
- Using the Query Handler

Overview of Server-Side Programs

Java servlets and CGI programs have different strengths and uses. The following list illustrates the differences between these server-side programs:

- Java servlets are written in Java, which is a full-featured programing language for creating network applications.
- CGI (Common Gateway Interface) programs can be written in C, Perl, or other programming languages. All CGI programs have a standard way of passing information between clients and servers.

Types of Server-Side Applications That Run on the Server

The iPlanet Web Server can run the following types of server-side applications to dynamically generate content:

- Java servlets
- CGI programs

The iPlanet Web Server can also run programs that extend or modify the behavior of the server itself. These programs, known as plug-ins, are written using the Netscape Server Application Programming Interface (NSAPI). For information about writing and installing plug-in programs, see the *NSAPI Programmer's Guide.*

How Server-Side Applications Are Installed on the Server

Each type of program is installed onto the server differently. The following list summarizes the procedures:

- For Java servlets, you can create and deploy web applications. For more information, see "What the Server Needs to Run Servlets and JSPs," on page 326.
- For CGI programs, you can configure your server to recognize all files with certain filename extensions, or all files in specified directories as CGI programs, or both. For more information, see "Installing CGI Programs," on page 333, "Installing Windows NT CGI Programs," on page 337, and "Installing Shell CGI Programs for Windows NT," on page 340.

These installation procedures are described in the following sections.

Java Servlets and JavaServer Pages (JSP)

This section discusses how to install and use Java Servlets and JavaServer Pages on iPlanet Web Server.

The following topics are described:

- Overview of Servlets and JavaServer Pages
- What the Server Needs to Run Servlets and JSPs

- Working with Web Applications
- Deploying Web Applications Using wdeploy
- Deploying and Editing Web Applications with the User Interface
- Deploying Servlets and JSPs Not in Web Applications
- Configuring JVM Attributes
- Deleting Version Files

Overview of Servlets and JavaServer Pages

iPlanet Web Server 6.0 supports the Servlet 2.2 API specification, which allows servlets and JSPs to be included in web applications.

A web application is a collection of servlets, JavaServer Pages, HTML documents, and other web resources which might include image files, compressed archives, and other data. A web application may be packaged into an archive (a WAR file) or exist in an open directory structure.

NOTE Servlet API version 2.2 is fully backward compatible with version 2.1, so all existing servlets will continue to work without modification or recompilation.

To develop servlets, use Sun Microsystems' Java Servlet API. For information about using the Java Servlet API, see the documentation provided by Sun Microsystems at:

`http://java.sun.com/products/servlet/index.html`

A JSP is a page, much like an HTML page, that can be viewed in a web browser. However, in addition to HTML tags, it can include a set of JSP tags and directives intermixed with Java code that extend the ability of the web page designer to incorporate dynamic content in a page. These additional features provide functionality such as displaying property values and using simple conditionals. iPlanet Web Server 6.0 supports the JavaServer Pages (JSP) 1.1 API specification.

For information about creating JSPs, see Sun Microsystem's JavaServer Pages web site at:

`http://java.sun.com/products/jsp/index.html`

For information about developing servlets and JSPs for use with iPlanet Web Server, see the *Programmer's Guide to Servlets*.

What the Server Needs to Run Servlets and JSPs

To enable servlets, select the Java tab in the Server manager, then select the Enable/Disable Servlets/JSP tab. Check the Enable Java Globally box to enable servlets for the entire server. Check the Enable Java for Class box to enable servlets for a single virtual server class. You cannot enable servlets for a class unless Java is globally enabled. By default, Java is globally enabled and enabled for each virtual server class.

To enable JSPs, first enable servlets. You must also include the `jsp-servlet` element with `enable="true"` in the `web-apps.xml` file and add `tools.jar` to the JVM classpath. For more information, see the *Programmer's Guide to Servlets*.

iPlanet Web Server includes the Java Runtime Environment (JRE) but not the Java Development Kit (JDK). The server can run servlets and precompiled JSPs using the JRE, but it needs the JDK to run uncompiled JSPs. If you want to develop JSPs, you must tell iPlanet Web Server to use a custom JDK.

iPlanet Web Server 6.0 requires you to use official versions of JDK, with different platforms requiring different versions, as summarized in the *Programmer's Guide to Servlets*.

Check the *iPlanet Web Server Installation and Migration Guide* and the latest *Release Notes* for updates on required JDK versions.

JDK 1.2 (and other JDK versions) are available from Sun Microsystems at:

`http://java.sun.com/products/jdk/1.2/`

You can specify the path to the JDK in either of the following ways:

- You can specify the path during the server installation process.

  When you install iPlanet Web Server, one of the dialog boxes in the installation process asks if you want to use a custom Java Development Kit (JDK), and if so, you can specify the path to it.

- You can specify it after the server is installed.

  To specify the path to the JDK, switch to the Web Server Administration Server, select the Global Settings tab, and use the Configure JRE/JDK Paths page, as described in "Configuring JRE/JDK Paths," on page 58. You can also change the path to the JDK in this page.

Whether you specify the path to the JDK during installation or later, the path is the folder in which you installed the JDK.

Working with Web Applications

The following sections describe how to deploy, edit, and delete web applications either manually by using the `wdeploy` command line utility, or through the user interface.

Using the web-apps.xml File

Before you deploy web applications, you must modify the `web-apps.xml` file, which is specific to iPlanet Web Server. Each virtual server has its own `web-apps.xml` file, which defines contexts for a set of web applications running in that virtual server. The context information includes a context path of the web application and other properties such as how it handles session management or authentication.

Once deployed, your web applications are enabled by default. To disable a deployed web application manually you would need to modify the web-app.xml file as follows:

```
<vs>
<web-app uri="/mywebapp" dir="/webappdir" enable = "false" >
</web-app>
```

If you inadvertently deploy or edit more than one web application with the same description, and one of them is disabled, the server will ignore `enable = “false”` and continue with default setting of `enable = “true”`.

For more information about the `web-apps.xml` file, see the *Programmer’s Guide to Servlets*.

Deploying Web Applications Using wdeploy

Before you can deploy a web application manually, you must make sure that the *server_root*`/bin/https/httpsadmin/bin` directory is in your path and that the `IWS_SERVER_HOME` environment variable is set to your *server_root* directory.

You can use the `wdeploy` utility at the command line to deploy a WAR file into a virtual server web application environment:

`wdeploy deploy -u` *uri_path* `-i` *instance* `-v` *vs_id* `[-d` *directory*`]` *war_file* `-n`

You can also delete a virtual server web application:

`wdeploy delete -u` *uri_path* `-i` *instance* `-v` *vs_id* `hard|soft -n`

You can also list the web application URIs and directories for a virtual server:

`wdeploy list -i` *instance* `-v` *vs_id*

The command parameters have the following meanings:

uri_path The URI prefix for the web application.

instance	The server instance name.
vs_id	The virtual server ID.
idirectory	(optional) The directory to which the application is deployed, or from which the application is deleted. If not specified for deployment, the application is deployed to the document root directory.
hard | soft	Specifies whether the directory and the `web-apps.xml` entry are deleted (`hard`) or just the `web-apps.xml` entry is deleted (`soft`).
war_file	The WAR file name

CAUTION If you deploy a web application and do not specify a *directory*, the application is deployed to the document root directory. If you then delete the application using the `hard` parameter, the document root directory will be deleted.

When you execute the `wdeploy deploy` command, three things happen:

- A web application with the given *uri_path* and *directory* gets added to the `web-apps.xml` file.
- The WAR file gets extracted at the target *directory*.
- The server is dynamically reconfigured to load the new web application.

For example:

```
wdeploy deploy -u /hello -i server.iplanet.com -v acme.com
-d /iws60/https-server.iplanet.com/acme.com/web-apps/hello
/iws60/plugins/servlets/examples/web-apps/HelloWorld/HelloWorld.war
```

This utility results in the following `web-apps.xml` entry:

```
<vs>
   <web-app uri="/hello"
   dir="/iws60/https-server.iplanet.com/acme.com/webapps/hello"/>
</vs>
```

The `/iws60/https-server.iplanet.com/acme.com/web-apps/hello` directory has the following contents:

```
colors
index.jsp
META-INF
WEB-INF/
   web.xml
   /classes/
      HelloWorldServlet.class
      HelloWorldServlet.java
      SnoopServlet.class
      SnoopServlet.java
```

Using -n in the wdeploy Command

In iPlanet Web Server 6.0 SP1, after deploying or deleting a web application, `wdeploy` dynamically reconfigures the server, causing the server to load or unload the web application that was deployed or deleted. Previously, you had to explicitly reconfigure the server in order for your changes to take effect by doing one of the following:

- Using the reconfigure script
- Restarting the server
- Clicking the Apply link in the Administration User Interface.

Now a successful `wdeploy` command will automatically be enabled to service requests for a new web application, or to stop servicing requests for a deleted web application.

The `-n` option prevents `wdeploy` from automatically sending the reconfigure command to the web server. Use the `-n` option in your command when deploying or undeploying multiple web applications (in a script for example), and you want to reconfigure the server only once after the last web application is deployed.

Accessing Deployed Web Applications

After you have deployed an application, you can access it from a browser as follows:

`http://`*vs_urlhost*`[:`*vs_port*`]/`*uri_path*`/[`*index_page*`]`

The parts of the URL have the following meanings:

vs_urlhost	One of the `urlhosts` values for the virtual server.
vs_port	(optional) Only needed if the virtual server uses a non-default port.
uri_path	The same one you used to deploy the application. This is also the context path.
index_page	(optional) The page in the application that end users are meant to access first.

For example:

```
http://acme.com:80/hello/index.jsp
```

or:

```
http://acme.com/hello/
```

Deploying and Editing Web Applications with the User Interface

As of iPlanet Web Server 6.0 SP1, you can now deploy, edit, delete, disable, and enable web applications for a specified virtual server.

Deploying Web Applications

You can access the Deploy Web Applications page by selecting Deploy Web Applications under the Web Applications tab of the Virtual Server Manager.

To deploy a web application, follow these steps:

1. Enter the Path to the web application file in the Web Applications File field. This is set to `web-apps.xml` for the default virtual server. The path can be an absolute path or a file name in the configuration directory of the instance. If the file does not exist there, it will be created.

2. Select Local Machine or Server Machine from the WAR File On drop-down list.

   Select Local Machine when uploading a WAR file to your server. Select Server Machine when the WAR file already resides there.

3. Enter the path on the local or server machine to the `WAR` file containing the web application in the field provided.

   On server machines enter the absolute path to the WAR file.

   On local machines you can browse the available paths. Clicking browse will bring up the File Upload window, allowing you to select the WAR file to upload to your server.

4. Enter the URI on the virtual server for the web application in the field provided.

5. Enter the absolute path to the directory on the server machine where the contents of the WAR file will be extracted from. If the directory does not exist, one will be created.
6. Click OK.
7. Click Apply.
8. Select Dynamic Reconfiguration for your web application to be deployed.

Editing Web Applications

You can edit, delete, disable, or enable an already deployed web application. Access the Edit Web Applications page by selecting Edit Web Applications under the Web Applications tab of the Virtual Server Manager.

To edit, delete, disable, or enable an already deployed web application, follow these steps:

1. Select the action you wish to perform from the drop-down list in the Action column next to the web application you are editing. Choose:
    - Edit to change the URI where the web application can be accessed.
    - Delete to delete the web application entry from the web applications file and delete the directory where the application is deployed.
    - Disable to make the web application inaccessible from the URI, but not delete it.
    - Enable to reactivate web applications that were previously disabled.

CAUTION Deleting a web application also deletes the directory the application is deployed in.

2. (Optional) Enter a new URI in the URI field if you are editing the web application.
3. Click OK.
4. Click Apply.
5. Select Dynamic Reconfiguration for your web application to be deployed.

Deploying Servlets and JSPs Not in Web Applications

You can deploy 4.*x* servlets and JSPs outside of web applications, but only in the default virtual server. For information, see the *Programmer's Guide to Servlets*.

Configuring JVM Attributes

You can configure attributes for the Java Virtual Machine (JVM) in the Configure JVM page of the Servlets tab in the Server Manager.

For more information on these options, see the *Programmer's Guide to Servlets*.

Deleting Version Files

The Delete Version Files page on the Java tab of the Server Manager allows you to delete the files that contain the version numbers for the JavaServer Pages class cache and the session data cache. This page has the following fields:

Clear Session Data

Deletes the SessionData directory, which stores persistent session information if the server uses the MMapSessionManager session manager.

Delete JSP ClassCache Files

Deletes the ClassCache directory, which caches information for JavaServer Pages (JSP). The default location of this directory is:

```
server_root/https-server_id/ClassCache/virtual_server_id/webapp_uri
/
```

When the server serves a JSP page, it creates a .java and a .class file associated with the JSP and stores them in the JSP class cache under the ClassCache directory.

The server uses two directories to cache information for JavaServer Pages (JSP) and servlets:

- `ClassCache`

  The server uses the following directory to cache information for JavaServer Pages (JSP):

  server_root/`https-`*server_id*/`ClassCache`/*virtual_server_id*/*webapp_uri*/

  When the server serves a JSP page, it creates a `.java` and a `.class` file associated with the JSP and stores them in the JSP class cache under the `ClassCache` directory.

- `SessionData`

  If the server uses the `MMappedSessionManager` session manager, it stores persistent session information in the `SessionData` directory.

Each cache has a `version` file containing a version number that the server uses to determine the structure of the directories and files in the caches. You can clean out the caches by simply deleting the version file.

When the server starts up, if it does not find the version files, it deletes the directory structure for the corresponding caches and re-creates the version files. Next time the server serves a JSP page, it recreates the JSP class cache. The next time the server serves a JSP page or servlet while using `MMappedSessionManager` session manager, it recreates the session data cache.

If a future upgrade of the server uses a different format for the caches, the server will check the number in the version file and clean up the caches if the version number is not correct.

Installing CGI Programs

This section discusses how to install CGI programs. The following topics are described:

- Overview of CGI
- Specifying a CGI Directory
- Specifying CGI as a File Type
- Downloading Executable Files

In addition, the following sections discuss how to install Windows NT-specific CGI programs:

- Installing Windows NT CGI Programs
- Installing Shell CGI Programs for Windows NT

Overview of CGI

Common Gateway Interface (CGI) programs can be defined with any number of programming languages. On a Unix/Linux machine, you’re likely to find CGI programs written as Bourne shell or Perl scripts.

NOTE Under Unix/Linux, there are extra `CGIStub` processes running that the server uses to aid in CGI execution. These processes are created only during the first access to a CGI. Their number varies depending upon the CGI load on the server. Do not kill these CGIStub processes. They disappear when the server is stopped.

On a Windows NT computer, you might find CGI programs written in C++ or batch files. For Windows NT, CGI programs written in a Windows-based programming language such as Visual Basic use a different mechanism to operate with the server. They are called Windows NT CGI programs. See "Installing Windows NT CGI Programs," on page 337 for information about Windows NT CGI.

NOTE In order to run the command-line utilities, you need to manually set the `Path` variable to include *server_root*`/bin/https/bin`.

Regardless of the programming language, all CGI programs accept and return data in the same manner. For information about writing CGI programs, see the following sources of information:

- *Programmer's Guide for iPlanet Web Server*
- *The Common Gateway Interface* at:

  `http://hoohoo.ncsa.uiuc.edu/cgi/overview.html`
- Articles about CGI available on the online documentation web site at:

  `http://www.iplanet.com/docs`

There are two ways to store CGI programs on your server machine:

- Specify a directory that contains only CGI programs. All files are run as programs regardless of the file extensions.
- Specify that CGI programs are all a certain file type. That is, they all use the file extensions `.cgi`, `.exe`, or `.bat`. The programs can be located in any directory in or under the document root directory.

You can enable both options at the same time if desired.

There are benefits to either implementation. If you want to allow only a specific set of users to add CGI programs, keep the CGI programs in specified directories and restrict access to those directories. If you want to allow anyone who can add HTML files to be able to add CGI programs, use the file type alternative. Users can keep their CGI files in the same directories as their HTML files.

If you choose the directory option, your server attempts to interpret any file in that directory as a CGI program. By the same token, if you choose the file type option, your server attempts to process any files with the file extensions `.cgi`, `.exe`, or `.bat` as CGI programs. If a file has one of these extensions but is not a CGI program, an error occurs when a user attempts to access it.

NOTE By default, the file extensions for CGI programs are `.cgi`, `.exe` and `.bat.` However, you can change which extensions indicate CGI programs by modifying the MIME types file. You can do this by choosing the Server Preferences tab and clicking the MIME Types link.

Specifying a CGI Directory

To specify a CGI-only directory for a class of virtual servers, perform the following steps:

1. From the Class Manager, choose the Programs tab.

   The CGI Directory window appears.

2. In the URL Prefix field, type the URL prefix to use for this directory. That is, the text you type appears as the directory for the CGI programs in URLs.

   For example, if you type `cgi-bin` as the URL prefix, then all URLs to these CGI programs have the following structure:

   `http://`*yourserver.domain.com*`/cgi-bin/`*program-name*

NOTE The URL prefix you specify can be different from the real CGI directory you specify in the previous step.

3. In the CGI Directory text field, type the location of the directory as an absolute path. Note that this directory doesn't have to be under your document root. This is the reason that you need to specify a URL prefix in the next step.

4. Click OK.

5. Save and apply your changes.

To remove an existing CGI directory, click that directory's Remove button in the CGI Directory form. To change the URL prefix or CGI directory of an existing directory, click that directory's Edit button.

Copy your CGI programs into the directories you've specified. Remember that any files in those directories will be processed as CGI files, so don't put HTML files in your CGI directory.

Configuring Unique CGI Attributes for Each Software Virtual Server

To specify CGI attributes for a single virtual server, perform the following steps:

1. From the Class Manager, choose the Manager Virtual Servers button.
2. From the Virtual Server Manager, choose the Settings tab.
3. In the CGI User text field, type the name of the user to execute CGI programs as.
4. In the CGI Group text field, type the name of the group to execute CGI programs as.
5. In the CGI Directory text field, type the directory to chdir to after chroot but before execution begins.
6. (Unix only) In the CGI Nice text field, type an increment that determines the CGI program's priority relative to the server. Typically, the server is run with a nice value of 0 and the nice increment would be between 0 (the CGI program runs at same priority as server) and 19 (the CGI program runs at much lower priority than server). While it is possible to increase the priority of the CGI program above that of the server by specifying a nice increment of -1, this is not recommended.
7. In the Chroot Directory text field, type the directory to chroot to before execution begins.
8. Click OK.
9. Save and apply your changes.

Specifying CGI as a File Type

To specify CGI programs as a file type, perform the following steps:

1. From the Class Manager, choose the Programs tab.
2. Click the CGI File Type page.

   The CGI as a File Type window appears.
3. From the Editing picker, choose the resource you want this change to apply to.
4. Click the Yes radio button under Activate CGI as a File Type.
5. Click OK.
6. Save and apply your changes.

The CGI files must have the file extensions `.bat`, `.exe`, or `.cgi`. Any non-CGI files with those extensions are processed by your server as CGI files, causing errors.

Downloading Executable Files

If you're using `.exe` as a CGI file type, you cannot download `.exe` files as executables.

One solution to this problem is to compress the executable files that you want users to be able to download, so that the extension is not `.exe`. This solution has the added benefit of making the download time shorter.

Another possible solution is to remove `.exe` as a file extension from the `magnus-internal/cgi` type and add it instead to the `application/octet-stream` type (the MIME type for normal downloadable files). You can do this through the Server Manager, by choosing the Server Preferences tab and clicking the MIME Types link. However, the disadvantage to this method is that after making this change you cannot use `.exe` files as CGI programs.

Another solution is to edit your server's `obj.conf` file to set up a download directory, where any file in the directory is downloaded automatically. The rest of the server won't be affected. For directions on setting up this directory, see the technical note at:

`http://help.netscape.com/kb/server/960513-130.html`

Installing Windows NT CGI Programs

This section discusses how to install Windows NT CGI Programs. The following topics are included in this section:

- Overview of Windows NT CGI Programs
- Specifying a Windows NT CGI Directory
- Specifying Windows NT CGI as a File Type

Overview of Windows NT CGI Programs

Windows NT CGI programs are handled much as other CGI programs. You specify a directory that contains only Windows NT CGI programs, or you specify that all Windows NT CGI programs have the same file extension. Note that like other CGI programs, you can use both methods at the same time if you want to. For example, you can create a directory for all your Windows NT CGI programs, and specify a Windows NT CGI file extension.

Although Windows NT CGI programs behave like regular CGI programs, your server processes the actual programs slightly differently. Therefore, you need to specify different directories for Windows NT CGI programs. If you enable the Windows NT CGI file type, it uses the file extension `.wcg`.

iPlanet Web Servers support the Windows NT CGI 1.3a informal specification, with the following differences:

- The following keywords have been added to the [CGI] section to support security methods:
  - HTTPS: its value is on or off, depending on whether the transaction is conducted through SSL.
  - HTTPS Keysize: when HTTPS is on, this value reports the number of bits in the session key used for encryption.
  - HTTPS Secret Keysize: when HTTPS is on, this value reports the number of bits used to generate the server’s private key.
- The keyword Document Root in the [CGI] section might not refer to the expected document root because the server does not have a single document root. The directory returned in this variable is the root directory for the Windows NT CGI program.
- The keyword Server Admin in the [CGI] section is not supported.
- The keyword Authentication Realm in the [CGI] section is not supported.
- Forms sent with multi-part/form-data encoding are not supported.

Specifying a Windows NT CGI Directory

To specify a Windows NT CGI-only directory:

1. From the Class Manager, choose the Programs tab.
2. Click the WinCGI Directory link.

   The WinCGI Directory window appears.
3. In the URL Prefix text field, enter the URL prefix you want to use for this directory.

   That is, the text you type appears as the directory for the Windows NT CGI programs in URLs. For example, if you type `wcgi-programs` as the URL prefix, then all URLs to these Windows NT CGI programs have the following structure:

   http://*yourserver*.domain.com/wcgi-programs/program-name

NOTE	The URL prefix you specify can be different from the real Windows NT CGI directory you specify in Step 5.

4. Choose whether you want to enable script tracing.

Click the Yes or No radio button under “Enable Script Tracing?”.

CGI parameters are passed from the server to Windows NT CGI programs through files, which the server normally deletes after the Windows NT CGI program finishes execution. If you enable script tracing, these files are retained in a `/temp` directory or wherever the environment variables TMP and TEMP are pointing. Also, any window that the Windows NT CGI program brings up is shown when script tracing is enabled.

5. In the WinCGI Directory field, enter the location of the directory as an absolute path.

Note that this directory doesn’t have to be under your document root. This is the reason that you need to specify a URL prefix in Step 3.

6. Click OK.

7. Save and apply your changes.

To remove an existing Windows NT CGI directory, click that directory’s Remove button in the Windows NT CGI Directory form. To change the URL prefix or Windows NT CGI directory of an existing directory, click that directory’s Edit button.

Copy your Windows NT CGI programs into the directories you’ve specified. Remember that any file in those directories is processed as a Windows NT CGI file.

Specifying Windows NT CGI as a File Type

To specify a file extension for Windows NT CGI files, perform the following steps:

1. From the Server Manager, choose the Server Preferences tab.

2. Click the MIME Types link.

The Global MIME Types window appears. For more information on the Global MIME Types, see “Choosing MIME Types,” on page 148.

3. Add a new MIME type with the following settings:

- Type: `type`
- Content type: `magnus-internal/wincgi`.
- File Suffix: Enter the file suffixes that you want the server to associate with Windows NT CGI. If you activated CGI, WinCGI, and shell CGI file types, you must specify a different suffix for each type of CGI. For example, you can’t use the suffix `.exe` for both a CGI program and a shell CGI program. If you need to, you can edit the other MIME type fields on the page so that the suffixes are unique.

4. Click the New Type button.

5. Save and apply your changes.

Installing Shell CGI Programs for Windows NT

This section discusses how to install Shell CGI Programs for Windows NT. The following topics are included in this section:

- Overview of Shell CGI Programs for Windows NT
- Specifying a Shell CGI Directory (Windows NT)
- Specifying Shell CGI as a File Type (Windows NT)

Overview of Shell CGI Programs for Windows NT

Shell CGI is a server configuration that lets you run CGI applications using the file associations set in Windows NT.

For example, if the server gets a request for a shell CGI file called `hello.pl`, the server uses the Windows NT file associations to run the file using the program associated with the `.pl` extension. If the `.pl` extension is associated with the program `C:\bin\perl.exe`, the server attempts to execute the `hello.pl` file as follows:

```
c:\bin\perl.exe hello.pl
```

The easiest way to configure shell CGI is to create a directory in your server's document root that contains only shell CGI files. However, you can also configure the server to associate specific file extensions with shell CGI by editing MIME types from the iPlanet Web Server.

NOTE For information on setting Windows NT file extensions, see your Windows NT documentation.

Specifying a Shell CGI Directory (Windows NT)

To create a directory for your shell CGI files, perform the following steps:

1. Create the shell directory on your computer. This directory doesn't have to be a subdirectory of your document root directory.
2. From the Server Manager, choose the Class Manager tab.
3. Next, choose the Programs tab.

   The shell CGI Directory link is highlighted and the CGI window appears.
4. In the URL Prefix field, enter the URL prefix you want to associate with your shell CGI directory.

   For example, suppose you store all shell CGI files in a directory called `C:/docs/programs/cgi/shell-cgi`, but you want users to see the directory as `http://www.`*yourserver*`.com/shell/`. In this case, you would type `shell` as the URL prefix.
5. In the Shell CGI Directory field, enter the absolute path to the directory you created.

CAUTION The server must have read and execute permissions to this directory. For Windows NT, the user account the server runs as (for example, `LocalSystem`) must have rights to read and execute programs in the shell CGI directory.

6. Make sure that any files in the shell CGI directory also have file associations set in Windows NT. The server returns an error if it attempts to run a file that has no file-extension association.

Specifying Shell CGI as a File Type (Windows NT)

You can use the iPlanet Web Server's MIME Types window to associate a file extension with the shell CGI feature. This is different from creating an association in Windows NT.

To associate a file extension with the shell CGI feature in the server, for example, you can create an association for files with the `.pl` extension. When the server gets a request for a file with that extension, the server knows to treat the file as a shell CGI file by calling the executable associated in Windows NT with that file extension.

To associate a file extension as a shell CGI file, perform the following steps:

1. Create the shell directory on your computer. This directory doesn't have to be a subdirectory of your document root directory.
2. From the Server Manager, choose Server Preferences.
3. Click the MIME Types link.

   The Global MIME Types window appears. For more information on the Global MIME Types, see "Choosing MIME Types," on page 148.
4. Add a new MIME type with these settings:
   - Type: `type`
   - Content type: `magnus-internal/shellcgi`.
   - File Suffix: Enter the file suffixes that you want the server to associate with shell CGI. If you activated CGI, WinCGI, and shell CGI file types, you must specify a different suffix for each type of CGI. For example, you can't use the suffix `.exe` for both a CGI program and a shell CGI program. If you need to, you can edit the other MIME type fields on the page so that the suffixes are unique.
5. Click the New Type button.
6. Save and apply your changes.

Using the Query Handler

NOTE The use of Query Handlers is outdated. Although iPlanet Web Server and Netscape Navigator clients still support it, it is rarely used. It is much more common for people to use forms in their HTML pages to submit queries.

You can specify a default query handler CGI program. A query handler processes text sent to it via the ISINDEX tag in an HTML file.

ISINDEX is similar to a form text field in that it creates a text field in the HTML page that can accept typed input. Unlike the information in a form text field, however, the information in the ISINDEX box is immediately submitted when the user presses Return. When you specify your default query handler, you tell your server to which program to direct the input. For an in-depth discussion of the ISINDEX tag, see an HTML reference manual.

To set a query handler, perform the following steps:

1. From the Server Manager, choose the Programs tab.
2. Click the Query Handler link.

   The Query Handler window appears.
3. Use the Editing Picker to select the resource you want to set with a default query handler.

   If you choose a directory, the query handler you specify runs only when the server receives a URL for that directory or any file in that directory.
4. In the Default Query Handler field, enter the full path for the CGI program you want to use as the default for the resource you chose.
5. Click OK.
6. Save and apply your changes.

Chapter 16

Content Management

This chapter describes how you can configure and manage content for classes of virtual servers and virtual servers.

This chapter contains the following sections:

- Setting the Primary Document Directory
- Setting Additional Document Directories
- Customizing User Public Information Directories (Unix/Linux)
- Restricting Symbolic Links (Unix/Linux)
- Enabling Remote File Manipulation
- Configuring Document Preferences
- Configuring URL Forwarding
- Customizing Error Responses
- Changing the Character Set
- Setting the Document Footer
- Using htaccess
- Setting up Server-Parsed HTML
- Setting Cache Control Directives
- Using Stronger Ciphers

Setting the Primary Document Directory

The primary document directory (also called the document root) is the central directory where you store all the files you want to make available to remote clients.

When you add a class, you specify a document directory with an absolute path. If you do not use a variable as part of that path, the document root for every virtual server in the class will default to the same directory. You can then change them individually in the Class Manager.

Another approach is to use a variable when you set the path for the class. For example, you can use the `$id` variable to create a directory named with the virtual server id for every virtual server in the class. You can set the class' document root to be *class_doc_root*/$id. Using this path, if your class' document directory is /`iplanet/servers/docs/$id`, the default document directory for a virtual server `vs1` that belongs to the class is `/iplanet/servers/docs/vs1`.

For more information about the document directory and how it is used at the server instance, class, and virtual server level, see"Document Root," on page 292.

To change the primary document directory to use a different path or variable, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Primary Document Directory.
3. Enter an absolute directory path or a variable, or a path and variable combination next to the virtual server.

   If you include the variable $id at the end of your document root absolute path, every virtual server has a default document root of *class_doc_root*/*virtual_server_ID*. For example, if your class' document directory is /`iplanet/servers/docs/$id`, the default document directory for a virtual server `vs1` that belongs to the class is `/iplanet/servers/docs/vs1`.

   For more information about variables, see "Using Variables," on page 296.
4. Click OK.

For more information, see the online help for the Primary Document Directory page.

NOTE Typically, each virtual server has its own primary document directory.

Setting Additional Document Directories

Most of the time, the documents for a virtual or server instance are in the primary document directory. Sometimes, though, you may want to serve documents from a directory outside of the document root. You can do this by setting additional document directories. By serving from a document directory outside of the document root, you can let someone manage a group of documents without giving them access to your primary document root.

If you set up an additional document directory without using variables, that directory will be set at the class level, and used by all virtual servers in the class.

If you want to set up additional document directories for individual virtual servers in the class, you must use variables so that the directory the URL prefix is mapped to is different for every virtual server.

To add an additional document directory, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Additional Document Directories.
3. Choose the URL prefix to map.

   Clients send this URL to the server when they want documents.
4. Specify the directory to map those URLs to.
5. If you want to, use an existing configuration style to specify how this directory should be configured.
6. Click OK.

To for more information, see the online help for the Additional Document Directories page.

By default, the server instance has several additional document directories. They have the following prefixes:

- /manual
- /servlet

You should restrict access to these directories so that users cannot write to them. A sample ACL would be:

```
deny (all) anyone;
allow (rxli) all;
allow (wd) privileged_user;
```

Customizing User Public Information Directories (Unix/Linux)

Sometimes users want to maintain their own web pages. You can configure public information directories that let all the users on a server create home pages and other documents without your intervention.

You can only set these up for the entire class. There's no way to customize them on a per virtual server basis.

With this system, clients can access your server with a certain URL that the server recognizes as a public information directory. For example, suppose you choose the prefix ~ and the directory `public_html`. If a request comes in for `http://www.iplanet.com/~jdoe/aboutjane.html`, the server recognizes that `~jdoe` refers to a users' public information directory. It looks up `jdoe` in the system's user database and finds Jane's home directory. The server then looks at `~/jdoe/public_html/aboutjane.html`.

To configure your server to use public directories, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click User Document Directories.
3. Choose a user URL prefix.

   The usual prefix is ~ because the tilde character is the standard Unix/Linux prefix for accessing a user's home directory.
4. Choose the subdirectory in the user's home directory where the server looks for HTML files.

   A typical directory is `public_html`.
5. Designate the password file.

   The server needs to know where to look for a file that lists users on your system. The server uses this file to determine valid user names and to find their home directories. If you use the system password file for this purpose, the server uses standard library calls to look up users. Alternatively, you can create another user file to look up users. You can specify that user file with an absolute path.

   Each line in the file should have this structure (the elements in the `/etc/passwd` file that aren't needed are indicated with `*`):

   ```
   username:*:*:groupid:*:homedir:*
   ```

6. Choose whether to load the password database at startup.

   For more information, see “Loading the Entire Password File on Startup,” on page 349.

7. Choose whether to apply a configuration style.

8. Click OK.

For more information, see the online help for the User Document Directories page.

Another way to give users separate directories is to create a URL mapping to a central directory that all of your users can modify.

Restricting Content Publication

In some situations a system administrator may want to restrict what user accounts are able to publish content via user document directories. To restrict a user’s publishing, add a trailing slash to the user’s home directory path in the `/etc/passwd file`:

```
jdoe::1234:1234:John Doe:/home/jdoe:/bin/sh
```

becomes:

```
jdoe::1234:1234:John Doe:/home/jdoe/:/bin/sh
```

After you make this modification, iPlanet Web Server will not serve pages from this user’s directory. The browser requesting the URI receives a “404 File Not Found” error and a 404 error will be logged to the web server access log. No error will be logged to the errors log.

If, at a later time, you decide to allow this user to publish content, remove the trailing slash from the `/etc/passwd` entry, then restart the web server.

Loading the Entire Password File on Startup

You also have the option of loading the entire password file on startup. If you choose this option, the server loads the password file into memory when it starts, making user lookups much faster. If you have a very large password file, however, this option can use too much memory.

Using Configuration Styles

You can apply a configuration style for the server to control access to directories from public information directories. This prevents users from creating symbolic links to information you do not want made public. For more information on configuration files, see Chapter 17, “Applying Configuration Styles.”

Enabling Remote File Manipulation

When you enable remote file manipulation, clients are able to upload files, delete files, create directories, remove directories, list the contents of a directory, and rename files on your server. The file `obj.conf` in the directory *`server_root`*`/https-`*`serve-id`*`/config` contains the commands that are activated when you enable remote file manipulation. By activating these commands, you allow remote browsers to change a server’s documents. You should use access control to restrict write access to these resources to prevent unauthorized tampering.

Note that enabling remote file manipulations should have no effect on using content management systems such as Microsoft Frontpage.

Unix/Linux: You must have the correct permissions for your files or this function will not work; that is, the document root user must be the same as the server user.

To enable remote file manipulation, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Remote File Manipulation.
3. Choose to activate remote file manipulation.
4. Click OK.

For more information, see the online help for the Remote File Manipulation page.

Configuring Document Preferences

You use the Document Preferences page to set document preferences. This section discusses these topics:

- Setting the Document Preferences
- Entering an Index Filename

- Selecting Directory Indexing
- Specifying a Server Home Page
- Specifying a Default MIME Type
- Parsing the Accept Language Header

These settings are all configured for the class, not individual virtual servers.

Setting the Document Preferences

To set the document preferences, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Document Preferences.
3. Choose the appropriate field values, as discussed in the following sections.
4. Click OK.

The preferences you can set are discussed more fully in the sections that follow. For additional information, see the online help for the Document Preferences page.

Entering an Index Filename

If a document name is not specified in the URL the server automatically displays the index file. The default index files are `index.html` and `home.html`. If more than one index file is specified, the server looks in the order in which the names appear in this field until one is found. For example, if your index filenames are `index.html` and `home.html`, the server looks for `index.html` and if it doesn't find it looks for `home.html`.

Selecting Directory Indexing

A document directory will probably have several subdirectories. For example, there might be a directory called `products`, another called `people`, and so on. It's often helpful to let clients access an overview (or index) of these directories.

The server indexes directories by searching the directory for an index file called `index.html` or `home.html`, which is a file you create and maintain as an overview of the directory's contents. For more information, see the previous section, "Entering an Index Filename" on page 351. You can specify any file as an index file for a directory by naming it one of these default names, which means you can also use a CGI program as an index if CGI is activated.

If an index file isn't found, the server generates an index file that lists all the files in the document root.

CAUTION If your server is outside the firewall, turn off directory indexing to ensure that your directory structure and filenames are not accessible.

Specifying a Server Home Page

When end users first access the server, the first file they see is usually called a home page. Usually, this file has general information about your server and links to other documents.

By default, the server finds the index file specified in the Index Filename field in the Document Preferences page and uses that for the home page. However, you can also specify a file to use as the home page.

Specifying a Default MIME Type

When a document is sent to a client, the server includes a section that identifies the document's type, so the client can present the document in the right way. However, sometimes the server can't determine the proper type for the document because the document's extension is not defined for the server. In those cases, a default value is sent.

The default is usually text/plain, but you should set it to the type of file most commonly stored on your server. Some common MIME types include the following:

- `text/plain`
- `text/richtext`
- `image/jpeg`
- `application/x-tar`
- `application/x-gzip`
- `text/html`
- `image/tiff`
- `image/gif`
- `application/postscript`
- `audio/basic`

Parsing the Accept Language Header

When clients contact a server using HTTP 1.1, they can send header information describing the languages they accept. You can configure your server to parse this language information.

For example, if you store documents in Japanese and English, you could choose to parse the accept language header. When clients that have Japanese as the accept language header contact the server, they receive the Japanese version of the page. When clients that have English as the accept language header contact the server, they receive the English version.

If you do not support multiple languages, you should not parse the accept language header.

For more information on using the accept language header, see the section “Using the Accept-language Header” on page 398.

Configuring URL Forwarding

URL forwarding allows you to redirect document requests to another server. Forwarding URLs or redirection is a method for the server to tell a user that a URL has changed (for example, because you have moved files to another directory or server). You can also use redirection to seamlessly send a person who requests a document on one server to a document on another server.

For example, if you forward http://www.iplanet.com/info/movies to a prefix film.iplanet.com, the URL http://www.iplanet.com/info/movies redirects to http://film.iplanet.com/info/movies.

You can use variables to map directories to new directories. For example, you can map /new to /$docroot/new. The mapping will go to the document root for the virtual server.

For more information about variables, see “Using Variables,” on page 296.

Sometimes you may want to redirect requests for all the documents in one sub-directory to a specific URL. For example, if you had to remove a directory because it was causing too much traffic, or because the documents were no longer to be served for any reason, you could direct a request for any one the documents to a page explaining why the documents were no longer available. For example, a prefix on /info/movies could be redirected to http://www.iplanet.com/explain.html.

To configure URL forwarding, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click URL Forwarding.

3. Type the URL prefix you want to redirect, and whether you want to redirect it to another prefix or to a static URL.
4. Click OK.

For more information see the online help for the URL Forwarding page.

Customizing Error Responses

You can specify a custom error response that sends a detailed message to clients when they encounter errors from your virtual server. You can specify a file to send or a CGI program to run.

For example, you can change the way the server behaves when it gets an error for a specific directory. If a client tries to connect to a part of your server protected by access control, you might return an error file with information on how to get an account.

Before you can enable a custom error response, you must create the HTML file to send or the CGI program to run in response to an error. After you do this, enable the response in the Class Manager.

To enable a customized error response, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Error Responses.
3. Choose Entire Server from the resource picker to apply your change to the whole class, or navigate to the document root for a specific virtual server, or to a specific directory or within a specific virtual server.
4. For each error code you want to change, specify the absolute path to the file or CGI that contains the error response.
5. Click OK.

For more information see the online help for the Error Responses page.

Changing the Character Set

The character set of a document is determined in part by the language it is written in. You can override a client’s default character set setting for a document, a set of documents, or a directory by selecting a resource and entering a character set for that resource.

Netscape Navigator can use the MIME type `charset` parameter in HTTP to change its character set. If the server includes this parameter in its response, Netscape Navigator changes its character set accordingly. Examples are:

- `Content-Type: text/html;charset=iso-8859-1`
- `Content-Type: text/html;charset=iso-2022-jp`

The following `charset` names recognized by Netscape Navigator are specified in RFC 1700 (except for the names that begin with `x-`):

- `us-ascii`
- `iso-2022-jp`
- `x-euc-jp`
- `iso-8859-1`
- `x-sjis`
- `x-mac-roman`

Additionally, the following aliases are recognized for `us-ascii`:

- `ansi_x3.4-1968`
- `ansi_x3.4-1986`
- `ascii`
- `us`
- `cp367`
- `iso-ir-6`
- `iso_646.irv:1991`
- `iso646-us`
- `ibm367`

The following aliases are recognized for `iso_8859-1`:

- `latin1`
- `iso_8859-1:1987`
- `ibm819`
- `iso_8859-1`
- iso-ir-100
- `cp819`

To change the character set, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click International Characters.

3. Choose Entire Server from the resource picker to apply your change to the whole class, or navigate to the document root for a specific virtual server, or to a specific directory or within a specific virtual server.

4. Set the character set for all or part of the server.

   If you leave this field blank, the character set is set to NONE.

5. Click OK.

For more information, see the online help for the International Characters page.

Setting the Document Footer

You can specify a document footer, which can include the last-modified time, for all the documents in a certain section of the server. This footer works for all files except output of CGI scripts or parsed HTML (.shtml) files. If you need your document footer to appear on CGI-script output or parsed HTML files, enter your footer text into a separate file and add a line of code or another server-side include to append that file to the page's output.

To set the document footer, follow these steps:

1. From the Class Manager, click the Content Management tab.

2. Click Document Footer.

3. Choose Entire Server from the resource picker to apply your change to the whole class, or navigate to the document root for a specific virtual server, or to a specific directory or within a specific virtual server.

   If you choose a directory, the document footer applies only when the server receives a URL for that directory or any file in that directory.

4. Specify the type of files that you want to have include the footer.

5. Specify the date format.

6. Type any text you want to have appear in the footer.

   The maximum number of characters for a document footer is 765. If you want to include the date the document was last modified, type the string :LASTMOD:.

For more information see the online help for the Document Footer page.

Using htaccess

For information on using htaccess, see “Using .htaccess Files,” on page 185.

Restricting Symbolic Links (Unix/Linux)

You can limit the use of the file system links in your server. File system links are references to files stored in other directories or file systems. The reference makes the remote file as accessible as if it were in the current directory. There are two types of file system links:

- **Hard links**—A hard link is really two filenames that point to the same set of data blocks; the original file and the link are identical. For this reason, hard links cannot be on different file systems.
- **Symbolic (soft) links**—A symbolic link consists of two files, an original file that contains the data, and another that points to the original file. Symbolic links are more flexible than hard links. Symbolic links can be used across different file systems and can be linked to directories.

For more information about hard and symbolic links, see your Unix/Linux system documentation.

File system links are an easy way to create pointers to documents outside of the primary document directory and anyone can create these links. For this reason you might be concerned that people might create pointers to sensitive files (for example, confidential documents or system password files).

To restrict symbolic links, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Symbolic Links.
3. Choose Entire Server from the resource picker to apply your change to the whole class, or navigate to the document root for a specific virtual server, or to a specific directory or within a specific virtual server.
4. Choose whether to enable soft and/or hard links and the directory to start from.
5. Click OK.

For more information, see the online help for the Symbolic Link page.

Setting up Server-Parsed HTML

HTML is normally sent to the client exactly as it exists on disk without any server intervention. However, the server can search HTML files for special commands (that is, it can parse the HTML) before sending documents. If you want the server to parse these files and insert request-specific information or files into documents, you must first enable HTML parsing.

To parse HTML, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Parse HTML.
3. Choose a resource for which the server will parse HTML.

   Choose Entire Server from the resource picker to apply your change to the whole class, or navigate to the document root for a specific virtual server, or to a specific directory or within a specific virtual server.

   If you choose a directory, the server will parse HTML only when the server receives a URL for that directory or any file in that directory.
4. Choose whether to activate server-parsed HTML.

   You can activate for HTML file s but not the exec tag, or for HTML files and the exec tag, which allows HTML files to execute other programs on the server.
5. Choose which files to parse.

   You can choose whether to parse only files with the .shtml extension, or all HTML files, which slows performance. If you are using Unix/Linux, you can also choose to parse Unix/Linux files with the execute permission turned on, though that can be unreliable.
6. Click OK.

For more information on setting your server to accept parsed HTML, see the online help for the Parse HTML page.

For more information on using server-parsed HTML, see the *Programmer’s Guide*.

Setting Cache Control Directives

Cache-control directives are a way for iPlanet Web Server to control what information is cached by a proxy server. Using cache-control directives, you override the default caching of the proxy to protect sensitive information from being cached, and perhaps retrieved later. For these directives to work, the proxy server must comply with HTTP 1.1.

For more information HTTP 1.1, see the Hypertext Transfer Protocol--HTTP/1.1 specification (RFC 2068) at:

http://www.ietf.org/

To set cache control directives, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Cache Control Directives
3. Fill in the fields. Valid values for the response directives are as follows:
    - **Public.** The response is cachable by any cache. This is the default.
    - **Private.** The response is only cachable by a private (non-shared) cache.
    - **No Cache.** The response must not be cached anywhere.
    - **No Store.** The cache must not store the request or response anywhere in nonvolatile storage.
    - **Must Revalidate.** The cache entry must be revalidated from the originating server.
    - **Maximum Age (sec).** The client does not accept a response that has an age greater than this age.
4. Click OK.

For more information see the online help for the Cache Control Directives page.

Using Stronger Ciphers

For information on setting stronger ciphers, see “Setting Stronger Ciphers,” on page 121.

Chapter 17

Applying Configuration Styles

Configuration styles are an easy way to apply a set of options to specific files or directories that your various virtual servers maintain. For example, you can create a configuration style that sets up access logging. When you apply that configuration style to the files and directories that you want to log, you don’t have to individually configure access logging for all the files and directories in your virtual server.

This chapter includes the following sections:

- Creating a Configuration Style
- Assigning a Configuration Style
- Listing Configuration Style Assignments
- Editing a Configuration Style
- Removing a Configuration Style

Creating a Configuration Style

To create a configuration style, perform the following steps:

1. Access the Class Manager.
2. Choose the Styles tab.
3. Click the New Style link.
4. Type the name you want to give the configuration style. Click OK.

   iPlanet Web Server displays the Edit a Style page.
5. From the drop-down list, choose a configuration style to edit and click Edit this Style.
6. From the list of links available, click the category you want to configure for your style.

You can configure the information listed in Table 17-1.

7. Fill out the form that appears, and click OK.
8. Repeat step 4 and step 5 to make any other configuration changes to the configuration style. Click OK.

When you choose a style to edit, your Resource Picker lists configuration styles instead of other resources. After you have finished editing a style, click OK and Save and Apply. The Resource Picker exits the styles mode. You can also choose to exit the styles mode by choosing Exit styles mode from the Resource Picker. For more information about the Resource Picker, see “Using the Resource Picker,” on page 40 of Chapter 1, “Introduction to iPlanet Web Server.”

Table 17-1 Configuration Style Categories

Category	Description
CGI file type	Allows you to activate CGI as a file type. For more information about CGIs, see “Installing CGI Programs,” on page 333 in Chapter 15, “Extending Your Server With Programs.”
Character Set	Allows you to change the character set for a resource. For more information about character sets, see “Changing the Character Set,” on page 354 in Chapter 16, “Content Management.”
Default Query Handler	Allows you to set a default query handler for a server resource. For more information about query handling, see “Using the Query Handler,” on page 342 in Chapter 15, “Extending Your Server With Programs.”
Document Footer	Allows you to add a document footer to a server resource.
Dynamic Configuration	Allows you to give people a subset of configuration options without giving them access to the Server Manager.
Error Responses	Allows you to customize the error responses that clients see when they encounter an error from your server.
Log preferences	Allows you to set preferences for access logs. For more information about log preferences, see “Setting Log Preferences,” on page 202 in Chapter 9, “Using Log Files.”
Remote file manipulation	Enables you to allow clients to upload files, delete files, create directories, remove directories, list the contents of a directory, and rename files on your server.
Require Stronger Security	Allows you to enforce stronger security requirements.

Table 17-1 Configuration Style Categories *(Continued)*

Category	**Description**
Restrict Access	Allows you to restrict access to the entire server or parts of it. For more information about access control, see Chapter 8, “Controlling Access to Your Server.”
Server Parsed HTML	Allows you to specify whether the server parses files before they are sent to the client.

For more information, see The Create a New Style Page in the online help.

Assigning a Configuration Style

Once you’ve created a configuration style, you can assign it to files or directories in your virtual server. You can specify either individual files and directories or wildcard patterns (such as `*.gif`).

To assign a configuration style, perform the following steps:

1. Access the Class Manager.
2. Choose the Styles tab.
3. Click the Assign Style link.
4. Enter the prefix of the URL to which you are applying this configuration style.

   If you choose a directory inside the document root, only enter the path after the document root. If you enter `/*` after the directory, you apply the configuration style to all of the directory’s contents.
5. Select the configuration style you want to apply.

   To remove any configuration style previously applied to the resource, apply the None configuration style. Click OK.

For more information, see The Assign a Style Page in the online help.

Listing Configuration Style Assignments

After you have created configuration styles and applied them to files or directories, you can get a list of the configuration styles and where you applied them.

To list the configuration style assignments, perform the following steps:

1. Access the Class Manager.
2. Choose the Styles tab.
3. Click the List Assignments link.

   iPlanet Web Server displays the List Assignments page, showing the configuration styles you applied to server resources.
4. To edit a configuration style assignment, click the Edit link next to the configuration style name.

For more information, see The List Assignments Page in the online help.

Editing a Configuration Style

To edit a configuration style, perform the following steps:

1. Access the Class Manager.
2. Choose the Styles tab.
3. Click the Edit Style link.
4. Select the configuration style you want to edit and click the “Edit this style” button.
5. From the list of links available, click the category you want to configure for your style.

   For more information on these categories, see the section “Creating a Configuration Style” on page 361.
6. Fill out the form that appears, and then click OK.
7. Repeat Step 4 and Step 5 to make any other changes to the configuration style. Click OK.

When you choose a style to edit, your Resource Picker lists configuration styles instead of other resources. After you have finished editing a style, click OK and Save and Apply. The Resource Picker exits the styles mode. You can also choose to exit the styles mode by choosing Exit styles mode from the Resource Picker. For more information about the Resource Picker, see “Using the Resource Picker,” on page 40 of Chapter 1, “Introduction to iPlanet Web Server.”

For more information, see The Edit a Style Page in the online help.

Removing a Configuration Style

Before removing a configuration style, remove assignments that had the configuration style applied to them. If you do not do this before removing the configuration style, you must manually edit the `obj.conf` file of your class of virtual server, searching for the configuration style in the file and replacing it with `None`. If you don't do this search and replace, anyone who accesses the files or directories that had the deleted configuration style applied will get a server misconfiguration error message.

To remove a configuration style, perform the following steps:

1. Access the Class Manager.
2. Choose the Styles tab.
3. Click List Assignments link.
4. Select Edit Style Assignment you want to remove.
5. Click Remove this Assignment.

For more information, see The Remove a Style Page in the online help.

Part 5

Appendixes

Appendix A

Command Line Utilities

This appendix provides instructions for using command line utilities in place of the user interface screens.

This appendix contains the following sections:

- Formatting LDIF Entries
- HttpServerAdmin (Virtual Server Administration)

Formatting LDIF Entries

LDIF consists of one or more directory entries separated by a blank line. Each LDIF entry consists of an optional entry ID, a required distinguished name, one or more object classes, and multiple attribute definitions.

For information about formatting LDIF entries, see the iPlanet Directory Server 5.0 *Command and File Reference* or the *Netscape Schema Reference, Directory Server 4.0* on `http://docs.iplanet.com/docs/manuals/directory.html`.

Modifying Database Entries Using ldapmodify

You use the `ldapmodify` command-line utility to modify entries in an existing Directory Server database. `ldapmodify` opens a connection to the specified server using the distinguished name and password you supply, and modifies the entries based on LDIF update statements contained in a specified file. Because `ldapmodify` uses LDIF update statements, `ldapmodify` can do everything that `ldapdelete` can do.

For more information about command-line utilities used with Directory Server database entries, see the iPlanet Directory Server 5.0 *Command and File Reference* or the *Netscape Schema Reference, Directory Server 4.0* on `http://docs.iplanet.com/docs/manuals/directory.html`.

HttpServerAdmin (Virtual Server Administration)

`HttpServerAdmin` is a command line utility that performs the same administrative functions as the virtual server user interface in the Server Manager and the Class Manager. If you prefer to set up your virtual servers using the command line interface, use `HttpServerAdmin`.

`HttpServerAdmin` is in *server_root*`/bin/https/httpadmin/bin`.

Before you can run `HttpServerAdmin`, you need to set the environment variable IWS_SERVER_HOME to the server root directory in your environment.

For example, on Unix/Linux systems:

```
setenv IWS_SERVER_HOME /usr/iplanet/servers
```

On Windows NT systems:

1. On the Control Panel, choose System.
2. Click the Environment tab.
3. Type IWS_SERVER_HOME in the Variable field and the path to your server root in the Value field.
4. Click Set.
5. Click OK.

NOTE In order to perform all commands, you need to have write permissions to the file `server.xml` where the virtual server information is stored.

HttpServerAdmin Syntax

The `HttpServerAdmin` syntax is as follows:

`HttpServerAdmin` *command_name command_options* `-d` *server_root* `-sinst` *http_instance*

You can get an online explanation of the command parameters by typing the following command:

```
./HttpServerAdmin -h
```

There are four possible values for the *command_name* parameter:

- control
- create
- delete
- list

Each command has its own set of command options. For more information, see the sections in this chapter that describe each command.

Regardless of the value of the command parameter, the parameters shown in Table A-1 can apply to all uses of the `HttpServerAdmin` command.

Table A-1 HttpServerAdmin Parameters

Parameter	Value
`-d` *server_root*	(required). This parameter designates the path to the server root (the location where the server is installed).
`-sinst` *http_instance*	(required). This parameter designates which instance `HttpServerAdmin` affects.

control Command

Use the `control` command to start, stop, and disable classes and virtual servers. If you do not specify a virtual server, the command starts, stops or disables every virtual server in the class.

Options

Use the options shown in Table A-2 with the `control` command to control classes and virtual servers.

Table A-2 Control command options

Options	Value
-start	Starts the specified virtual server, or all virtual servers in the class if no virtual server is specified.
-stop	Stops the specified virtual server, or all virtual servers in the class if no virtual server is specified.
-disable	Disables the specified virtual server, or all virtual servers in the class if no virtual server is specified.

Syntax

`HttpServerAdmin control -cl` *classname*`, -`*control_option* `[-id` *virtual_server*`] -d` *server_root* `-sinst` *http_instance*

Parameters

Use these parameters with the command options to control virtual servers

Table A-3 Control command parameters

Parameters	Value
-cl *classname*	Designates the virtual server class
-id *virtual_server*	(optional) Designates the virtual server ID you are controlling.

Examples

```
HttpServerAdmin control -cl myclass -start -id myvirtualserver -d
/usr/iplanet/servers -sinst https-iplanet.com

HttpServerAdmin control -cl myclass -stop -id myvirtualserver -d
/usr/iplanet/servers -sinst https-iplanet.com

HttpServerAdmin control -cl myclass -disable -id myvirtualserver
-d /usr/iplanet/servers -sinst https-iplanet.com
```

create Command

Use the `create` command to create classes of virtual servers, virtual servers, listen sockets and connection groups.

Options

Use the options shown in Table A-4 with the `create` command to create classes, listen sockets, and virtual servers.

Table A-4 Create command options

Option	Value
-c	Creates a virtual server class.
-g	Creates a connection group.
-l	Creates a listen socket.
-v	Creates a virtual server.

Each of these options in turn has its own parameters, which are shown in the following sections.

Create Virtual Server Class

Use this option of the create command to create a virtual server class.

Syntax

`HttpServerAdmin create -c -cl` *classname* `[-docroot` *document_root*`] [-obj` *obj.conf_file*`] [-acptlang` *accept_language*`] -d` *server_root* `-sinst` *http_instance*

Parameters

Use the parameters shown in Table A-5 with the `create -c` command option to create classes.

Table A-5 Create virtual server class parameter

Parameter	Value
-cl *classname*	The name of the class you want to create.
-docroot *document_root*	(optional) The document root for the class. This has to be an absolute path.

Table A-5 Create virtual server class parameter

Parameter	**Value**
-obj *obj.conf_file*	(optional) The obj.conf file for the class. If you do not specify this parameter, the server creates the obj.conf file as *classname*.obj.conf. If you want a different name for the class' obj.conf file, specify it here.
-acptlang *accept_language*	(optional) If you do not specify this parameter, acptlang will be off by default.

Example

```
HttpServerAdmin create -c -cl myclass1 -d /export/iplanet/servers
-sinst https-iplanet.com
```

Create Connection Group

Use this option of the create command to create a connection group.

Syntax

`HttpServerAdmin create -g` *group_ID* `-lsid` *listen_socket* `-ip` *IPaddress* `-sname` *server_name* `-defaultvs` *default_virtual_server* `-d` *server_root* `-sinst` *`http_instance`*

Parameters

Use the parameters shown in Table A-8 with the `create -g` command option to create connection groups.

Table A-6 Create connection group parameters

Parameter	**Value**
-g *connection_group*	The ID of the connection group you are creating.
`-lsid` *listen_socket*	The ID of the listen socket you want associated with this connection group.
-ip *IP_address*	The IP address associated with this connection group.
-sname *server_name*	The server name.
-defaultvs *default_virtual_server*	The default virtual server the connection group connects to if the requested URL host cannot be found.

Examples

```
HttpServerAdmin create -g conngroup2 -lsid ls1 -ip 1.1.1.1 -sname
iplanet -defaultvs vs2 -d server_root -sinst https-iplanet.com
```

Create Listen Socket

Use this option of the create command to create a listen socket.

Syntax

`HttpServerAdmin create -l -ip` *ip_address* `-port` *port_number* `-sname` *server_name* `-defaultvs` *default_virtual_server* `[-sec security] [-acct` *number_of_accept_threads*`] -d` *server_root* `-sinst http_instance`

Parameters

Use the parameters shown in Table A-7 with the `create -l` command option to create listen sockets.

Table A-7 Create listen socket parameters

Parameter	**Value**
-ip *ip_address*	The IP address for the listen socket.
-port *port_number*	The port number for the listen socket.
-sname *server_name*	The server name to associate with the listen socket.
-defaultvs *default_virtual_server*	The ID of the default virtual server. This virtual server must exist before you can use it to create a listen socket.
-acct *number_of_accept_threads*	(optional) The number of accept threads for the listen socket.
-sec *on*	(optional) If specified, use on to enable security for the listen socket. If not specified, security is not enabled.

Example

```
HttpServerAdmin create -l -id ls3 -ip 0.0.0.0 -port 1333 -sname
austen -defaultvs vs2 -sec on -acct 4 -d /export/carey/server6
-sinst https-austen.com
```

Create Virtual Server

Use this option of the create command to create a virtual server.

Please note that if you do not include values for some of the optional parameters, defaults are provided. You can always change the default values after the virtual server is created.

Syntax

```
HttpServerAdmin create -v -id virtual_server -cl classname -urlh urlhosts
-conngroupid connection_group_ID[-state state][-docroot document_root] [-mime
mime_types_file] [-aclid acl_ID] -d server_root -sinst http_instance
```

Parameters

Use the parameters shown in Table A-8 with the `create -v` command option to create virtual servers.

Table A-8 Create listen socket parameters

Parameter	Value
-id *virtual_server*	The ID of the virtual server you are creating.
-cl *classname*	The class of which the virtual server will be a member.
-urlh *URL_hosts*	The URL hosts for the virtual server. You can specify more than one URL host, separated by a comma.
-conngroupid *connection_group_ID*	The connection group for the listen socket.
-state *state*	(optional) Valid values are on, off, and disable.
-docroot *document_root*	(optional) If you want to specify a document root for a virtual server, use this parameter. You must use an absolute path name.
-mime *mime_types_file*	(optional) The name of the MIME types file for the virtual server.
-aclid *acl_ID*	(optional) The ACL file ID <ACLID> used in the `server.xml` file

Examples

```
HttpServerAdmin create -v -id vs3 -cl class1 -urlh annh
-conngroupid group1 -d /export/iplanet/server6 -sinst
https-iplanet.com

HttpServerAdmin create -v -id vs4 -cl class1 -urlh annh,annh2
-conngroupid group1 -state off -mime mime.types -d
/export/iplanet/server6 -sinst https-iplanet.com
```

delete Command

Use the delete command to delete classes of virtual servers, virtual servers, and listen sockets.

Options

Use the options shown in Table A-9 with the `delete` command to delete classes.

Table A-9 Delete command options

Option	Value
-c	Deletes the specified virtual server class.
-g	Deletes the specified connection group.
-l	Deletes the specified listen socket IDs
-v	Deletes the specified virtual servers.

Delete Class

Use this option of the delete command to delete a virtual server class.

Syntax

`HttpServerAdmin delete -c -cl` *classname* `-d` *server_root* `-sinst` *http_instance*

Parameters

Use the parameters shown in Table A-9 with the `delete` command to delete classes.

Table A-10 Delete class parameters

parameter	Value
-c l *class*	The class name you want to delete.

Example

```
HttpServerAdmin delete -c -cl class1 -d /export/iplanet/server6
-sinst https-iplanet.com
```

Delete Connection Group

Use this option of the delete command to delete a connection group.

Syntax

`HttpServerAdmin delete -g -id` *connection_group* `-lsid` *listen_socket* `-d` *server_root* `-sinst` *http_instance*

Parameters

Use the parameters shown in Table A-9 with the `delete` command to delete a connection group.

Table A-11 Delete connection group parameters

parameter	Value
-id *connection_group*	The virtual server ID you want to delete
-lsid *listen_socket*	The listen socket ID the connection group belongs to.

Example

```
HttpServerAdmin delete -g -id conngroup3 -lsid ls2 -d
/export/iplanet/server6 -sinst https-iplanet.com
```

Delete Listen Socket

Use this option of the delete command to delete a listen socket.

Syntax

`HttpServerAdmin delete -l -id` *listen_socket* `-d` *server_root* `-sinst` *http_instance*

Parameters

Use the parameters shown in Table A-9 with the `delete` command to delete classes.

Table A-12 Delete class parameters

parameter	**Value**
-id *listen_socket*	The ID of the listen socket you want to delete.

Example

```
HttpServerAdmin delete -l -id ls3 -d /export/iplanet/server6
-sinst https-iplanet.com
```

Delete Virtual Server

Use this option of the delete command to delete a virtual server.

Syntax

`HttpServerAdmin delete -v -id` *virtual_server* `-cl` *classname* `-d` *server_root*
`-sinst` *http_instance*

Parameters

Use the parameters shown in Table A-9 with the `delete` command to delete a virtual server.

Table A-13 Delete virtual server parameters

parameter	**Value**
-id *virtual_server*	The virtual server ID you want to delete
-cl *class*	The class the virtual server belongs to.

Example

```
HttpServerAdmin delete -v -id vs3 -cl class1 -d
/export/iplanet/server6 -sinst https-iplanet.com
```

list Command

Use the `list` command to list classes of virtual servers, virtual servers, listen sockets and connection groups.

Syntax

`HttpServerAdmin list -`*command_option* `-d` *server_root* `-sinst` *http_instance*

Options

Table A-14 List command options

Option	**Value**
-c	Lists all virtual server classes.
-g -lsid *listen_socket*	Lists all connection groups for a listen socket.
-l	Lists all listen sockets.
-v	Lists all virtual servers.

Example

```
HttpServerAdmin list -c -d /export/iplanet/server6 -sinst
https-iplanet.com

HttpServerAdmin list -l -d /export/iplanet/server6 -sinst
https-iplanet.com

HttpServerAdmin list -g -lsid ls1 -d /export/iplanet/server6
-sinst https-iplanet.com
```

The list of information appears in your command window.

Appendix B

HyperText Transfer Protocol

This appendix provides a short introduction to a few HyperText Transfer Protocol (HTTP) basics. For more information on HTTP, see the Internet Engineering Task Force (IETF) home page at:

```
http://www.ietf.org/home.html
```

This appendix contains the following sections:

- About HyperText Transfer Protocol (HTTP)
- Requests
- Responses

About HyperText Transfer Protocol (HTTP)

The **HyperText Transfer Protocol (HTTP)** is a protocol (a set of rules that describe how information is exchanged on a network) that allows a web browser and a web server to “talk” to each other using the ISO Latin1 alphabet, which is ASCII with extensions for European languages.

HTTP is based on a request/response model. The client connects to the server and sends a request to the server. The request contains the following: request method, URI, and protocol version. The client then sends some header information. The server’s response includes the return of the protocol version, status code, followed by a header that contains server information, and then the requested data. The connection is then closed.

The iPlanet Web Server 4.x supports HTTP 1.1. Previous versions of the server supported HTTP 1.0. The server is conditionally compliant with the HTTP 1.1 proposed standard, as approved by the Internet Engineering Steering Group (IESG) and the Internet Engineering Task Force (IETF) HTTP working group. For more information on the criteria for being conditionally compliant, see the Hypertext Transfer Protocol—HTTP/1.1 specification (RFC 2068) at:

```
http://www.ietf.org/html.charters/http-charter.html
```

Requests

A request from a client to a server includes the following information:

- Request method
- Request header
- Request data

Request Method

A client can request information using a number of methods. The commonly used methods include the following:

- `GET`—Requests the specified document
- `HEAD`—Requests only the header information for the document
- `POST`—Requests that the server accept some data from the client, such as form input for a CGI program
- `PUT`—Replaces the contents of a server's document with data from the client

Request Header

The client can send header fields to the server. Most are optional. Some commonly used request headers are shown in Table B-1.

Table B-1 Common request headers

Request header	**Description**
`Accept`	The file types the client can accept.

Table B-1 Common request headers *(Continued)*

Request header	Description
`Authorization`	Used if the client wants to authenticate itself with a server; information such as the username and password are included.
`User-agent`	The name and version of the client software.
`Referer`	The URL of the document where the user clicked on the link.
`Host`	The Internet host and port number of the resource being requested.

Request Data

If the client has made a `POST` or `PUT` request, it can send data after the request header and a blank line. If the client sends a `GET` or `HEAD` request, there is no data to send; the client waits for the server's response.

Responses

The server's response includes the following:

- Status code
- Response header
- Response data

Status Code

When a client makes a request, one item the server sends back is a status code, which is a three-digit numeric code. There are four categories of status codes:

- Status codes in the 100–199 range indicate a provisional response.
- Status codes in the 200–299 range indicate a successful transaction.
- Status codes in the 300–399 range are returned when the URL can't be retrieved because the requested document has moved.
- Status codes in the 400–499 range indicate the client has an error.
- Status codes of 500 and higher indicate that the server can't perform the request, or an error has occurred.

Table B-2 contains some common status codes.

Table B-2 Common HTTP status codes

Status code	Meaning
200	OK; successful transmission. This is not an error.
302	Found. Redirection to a new URL. The original URL has moved. This is not an error; most browsers will get the new page.
304	Use a local copy. If a browser already has a page in its cache, and the page is requested again, some browsers (such as Netscape Navigator) relay to the web server the “last-modified” timestamp on the browser’s cached copy. If the copy on the server is not newer than the browser’s copy, the server returns a 304 code instead of returning the page, reducing unnecessary network traffic. This is not an error.
401	Unauthorized. The user requested a document but didn’t provide a valid username or password.
403	Forbidden. Access to this URL is forbidden.
404	Not found. The document requested isn’t on the server. This code can also be sent if the server has been told to protect the document by telling unauthorized people that it doesn’t exist.
500	Server error. A server-related error occurred. The server administrator should check the server’s error log to see what happened.

Response Header

The response header contains information about the server and information about the document that will follow. Common response headers are shown in Table B-3.

Table B-3 Common response headers

Response header	Description
Server	The name and version of the web server.
Date	The current date (in Greenwich Mean Time).
Last-modified	The date when the document was last modified.
Expires	The date when the document expires.

Table B-3 Common response headers

Response header	Description
`Content-length`	The length of the data that follows (in bytes).
`Content-type`	The MIME type of the following data.
`WWW-authenticate`	Used during authentication and includes information that tells the client software what is necessary for authentication (such as username and password).

Response Data

The server sends a blank line after the last header field. The server then sends the document data.

Appendix C

ACL File Syntax

This appendix describes the access-control list (ACL) files and their syntax. ACL files are text files that contain lists that define who can access resources stored on your web server. By default, the web server uses one ACL file that contains all of the lists for access to your server. However, you can create multiple ACL files and reference them in the `obj.conf` file.

You need to know the syntax and function of ACL files if you plan on customizing access control using the access-control API. For example, you might use the access control API to interface with another database, such as an Oracle or Informix database. For more information on the API, see the iPlanet documentation site at:

```
http://www.iplanet.com/docs
```

This appendix contains the following sections:

- ACL File Syntax
- Referencing ACL Files in obj.conf

ACL File Syntax

All ACL files must follow a specific format and syntax. An ACL file is a text file containing one or more ACLs. All ACL files must begin with the version number they use. There can be only one version line and it can appear after any comment lines. iPlanet Web Server 6.0 uses version 3.0. For example:

```
version 3.0;
```

You can include comments in the file by beginning the comment line with the # sign.

Each ACL in the file begins with a statement that defines its type. ACLs can follow one of three types:

- **Path ACLs** specify an absolute path to the resource they affect
- **URI (Uniform Resource Indicator) ACLs** specify a directory or file relative to the server's document root.
- **Named ACLs** specify a name that is referenced in resources in the `obj.conf` file. The server comes with a "default" named resource that allows read access to anyone and write access to users in the LDAP directory. Even though you can create a named ACL from the iPlanet Web Server windows, you must manually reference the named ACLs with resources in the `obj.conf` file.

Path and URI ACLs can include wildcards at then end of the entry. For example: `/a/b/*.` Wildcards placed anywhere except at the end of the entry will not work.

The type line begins with the letters `acl` and then includes the type information in double-quotation marks followed by a semicolon. Each type information for all ACLs must be a unique name--even among different ACL files. The following lines are examples of several different types of ACLs:

```
acl "path=C:/iPlanet/Servers/docs/mydocs/";
acl "default";
acl "uri=/mydocs/";
```

After you define the type of ACL, you can have one or more statements that define the method used with the ACL (authentication statements) and the people and computers who are allowed or denied access (authorization statements). The following sections describe the syntax for these statements.

This section includes the following topics:

- Authentication Methods
- Authorization Statements
- The Default ACL File

Authentication Methods

ACLs can optionally specify the authentication method the server must use when processing the ACL. There are three general methods:

- Basic (default)
- Digest

- SSL

Basic and digest require users to enter a username and password before accessing a resource.

SSL requires the user to have a client certificate. The web server must have encryption turned on, and the user’s certificate issuer must be in the list of trusted CAs to be authenticated.

By default, the server uses the Basic method for any ACL that doesn’t specify a method. Your server’s authentication database must be able to handle digest authentication sent by a user.

Each authenticate line must specify what attribute (users, groups, or both users and groups) the server authenticate. The following authentication statement, which would appear after the ACL type line, specifies basic authentication with users matched to individual users in the database or directory:

```
authenticate (user) {
     method = “basic”;
};
```

The following example uses SSL as the authentication method for users and groups:

```
authenticate (user, group) {
     method = “ssl”;
};
```

The following example allows any user whose username begins with the letters `sales`:

```
authenticate (user)

allow (all)

   user = sales*
```

If the last line was changed to `group = sales`, then the ACL would fail because the group attribute is not authenticated.

Authorization Statements

Each ACL entry can include one or more authorization statements. Authorization statements specify who is allowed or denied access to a server resource. Use the following syntax when writing authorization statements:

```
allow|deny [absolute] (right[,right...]) attribute expression;
```

Start each line with either allow or deny. It’s usually a good idea to deny access to everyone in the first rule and then specifically allow access for users, groups, or computers in subsequent rules. This is because of the hierarchy of rules. That is, if you allow anyone access to a directory called `/my_stuff`, and then you have a subdirectory `/my_stuff/personal` that allows access to a few users, the access control on the subdirectory won’t work because anyone allowed access to the `/my_stuff` directory will also be allowed access to the `/my_stuff/personal` directory. To prevent this, create a rule for the subdirectory that first denies access to anyone and then allows it for the few users who need access.

However, in some cases if you set the default ACL to deny access to everyone, then your other ACL rules don’t need a “deny all” rule.

The following line denies access to everyone:

```
deny (all)
    user = "anyone";
```

This section includes the following topics:

- Hierarchy of Authorization Statements
- Attribute Expressions
- Operators For Expressions

Hierarchy of Authorization Statements

ACLs have a hierarchy that depends on the resource. For example, if the server receives a request for the document (URI) `/my_stuff/web/presentation.html`, the server builds a list of ACLs that apply for this URI. The server first adds ACLs listed in ‘check-acl’ statements of it’s `obj.conf` file. Then the server appends matching URI and PATH ACLs.

The server processes this list in the same order. Unless ‘absolute’ ACL statements are present, all statements are evaluated in order. If an ‘absolute allow’ or ‘absolute deny’ statement evaluates to ‘true’, the server stops processing and accepts this result.

If there are more than one ACLs that match, the server uses the last statement that matches. However, if you use an absolute statement, then the server stops looking for other matches and uses the ACL containing the absolute statement. If you have two absolute statements for the same resource, the server uses the first one in the file and stops looking for other resources that match.

```
version 3.0;
acl "default";
authenticate (user,group) {
     prompt="Web Server";
};
allow (read,execute,list,info)
     user = "anyone";
allow (write,delete)
     user = "all";
acl "uri=/my_stuff/web/presentation.html";
deny (all)
     user = "anyone";
allow (all)
     user = "joe";
```

Attribute Expressions

Attribute expressions define who is allowed or denied access based on their username, group name, host name, or IP address. The following lines are examples of allowing access to different people or computers:

- `user = “anyone”`
- `user = “smith*”`
- `group = “sales”`
- `dns = “*.iplanet.com”`
- `dns = “*.iplanet.com,*.mozilla.com”`
- `ip = “198.*”`
- `ciphers = “rc4”`
- `ssl = “on”`

You can also restrict access to your server by time of day (based on the local time on the server) by using the `timeofday` attribute. For example, you can use the `timeofday` attribute to restrict access to certain users during specific hours.

NOTE Use 24-hour time to specify times. For example, use 0400 to specify 4:00 a.m. or 2230 for 10:30 p.m.

The following example restricts access to a group of users called guests between 8:00 a.m. and 4:59 p.m:

```
allow (read)

    (group="guests") and
    (timeofday<0800 or timeofday=1700);
```

You can also restrict access by day of the week. Use the following three-letter abbreviations to specify days of the week: Sun, Mon, Tue, Wed, Thu, Fri, and Sat.

The following statement allows access for users in the premium group any day and any time. Users in the discount group get access all day on weekends and on weekdays anytime except 8am-4:59pm.

```
allow (read) (group="discount" and dayofweek="Sat,Sun") or
(group="discount" and (dayofweek="mon,tue,wed,thu,fri" and
(timeofday<0800 or timeofday=1700)))
or
(group="premium");
```

Operators For Expressions

You can use various operators in attribute expressions. Parentheses delineate the operator order of precedence. With `user`, `group`, `dns`, and `ip`, you can use the following operators:

- and
- `or`
- not
- = (equals)
- `!=` (not equal to)

With `timeofday` and `dayofweek`, you can use:

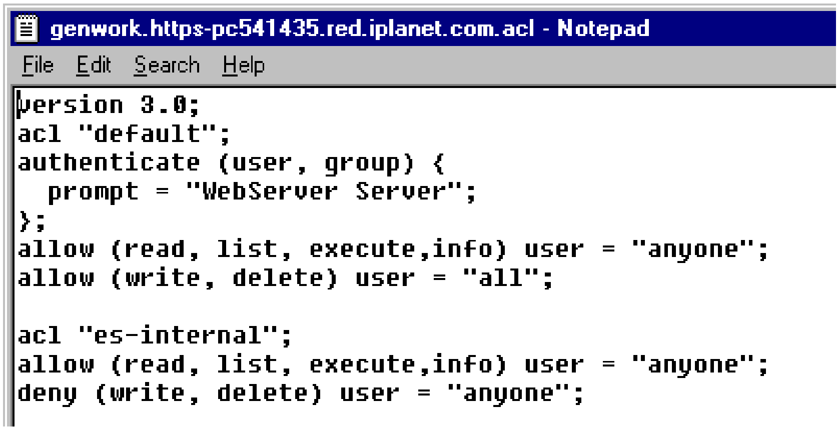
- greater than
- < less than

- = greater than or equal to
- <= less than or equal to

The Default ACL File

After installation, the *server_root*/`httpacl/generated.https-`*serverid*`.acl` file provided default settings for the server. The server uses the working file `genwork.https-`*serverid*`.acl` until you create settings in the user interface. When editing an ACL file, you could make changes in the `genwork` file, then save and apply the changes using iPlanet Web Server.

Figure C-1 genwork File.



```
version 3.0;
acl "default";
authenticate (user, group) {
  prompt = "WebServer Server";
};
allow (read, list, execute,info) user = "anyone";
allow (write, delete) user = "all";

acl "es-internal";
allow (read, list, execute,info) user = "anyone";
deny (write, delete) user = "anyone";
```

General Syntax Items

Input strings can contain the following characters:

- Letters a through z
- Numbers 0 through 9
- Period and underscore

If you use any other characters, you need to use double-quotation marks around the characters.

A single statement can be placed on its own line and be terminated with a semicolon. Multiple statements are placed within braces. A list of items must be separated by commas and enclosed in double-quotation marks.

Referencing ACL Files in obj.conf

If you have named ACLs or separate ACL files, you can reference them in the obj.conf file. You do this in the PathCheck directive using the check-acl function. The line has the following syntax:

```
PathCheck fn="check-acl" acl="aclname"
```

The aclname is a unique name of an ACL as it appears in any ACL file.

For example, you might add the following lines to your obj.conf file if you want to restrict access to a directory using the ACL named testacl:

```
<Object ppath="/usr/ns-home/docs/test/*"
PathCheck fn="check-acl" acl="testacl"
</Object
```

In the previous example, the first line is the object that states which server resource you want to restrict access to. The second line is the PathCheck directive that uses the check-acl function to bind the name ACL (testacl) to the object in which the directive appears. The testacl ACL can appear in any ACL file referenced in magnus.conf.

Appendix D

Internationalized iPlanet Web Server

The internationalized version of the iPlanet Web Server contains special features tailored for the non-U.S. environment. These features include a choice of user-interface language (Japanese, French, or German) and a choice of search engines that allow you to use text search on a variety of languages.

This appendix contains the following sections:

- General Information
- Search Information
- Servlet Internationalization

General Information

The following information covers the international considerations for general server capabilities:

- Installing the Server
- Entering UTF-8 Data
- Using the Accept-language Header
- Using Other Language Settings

Installing the Server

When you install the server, you choose what user-interface language to use, as well as what search engines to install.

For information on installing the international version of the server, see the iPlanet Web Server, Edition 6.0 *Release Notes*. You can access the *Release Notes* online via the link provided in the README file.

Entering UTF-8 Data

If you want to enter UTF-8 data on the Server Manager or the Administration Server pages, you need to be aware of the following issues:

File or Directory Names

If a file or directory name is to appear in a URL, it cannot contain 8-bit or multi-byte characters.

LDAP Users and Groups

For email addresses, use only those characters permitted in RFC 1700 (ftp://ds.internic.net/rfc/rfc1700.txt). User ID and password information must be stored in ASCII.

To make sure you enter characters in the correct format for users and groups, use a UTF-8 form-capable client (such as Netscape Communicator) to input 8-bit or multi-byte data.

If you let users access their own user and group information, they will need to use a UTF-8 form-capable client.

Using the Accept-language Header

When clients contact a server using HTTP, they can send header information that describes the various languages they accept. You can configure your server to parse this language information as described in Chapter 16, “Content Management”.

You can enable or disable the server to the `acceptlanguage` directive in the `server.xml` file.

Figure D-1 International Settings in server.xml

`acceptlanguage`	`on`, `off`	Enables or disables the `Accept-language` header parsing.

For example, if `acceptlanguage` is set to `on`, and a client sends the Accept-language header with the value `fr-CH,de`, when requesting the following URL:

`http://www.someplace.com/somepage.html`

Your server searches for the file in the following order:

1. The `Accept-language` list `fr-CH,de`.

   `http://www.someplace.com/fr_ch/somepage.html`

   `http://www.someplace.com/somepage_fr_ch.html`

   `http://www.someplace.com/de/somepage.html`

   `http://www.someplace.com/somepage_de.html`

2. Language codes without the country codes (`fr` in the case of `fr-CH`):

   `http://www.someplace.com/fr/somepage.html`

   `http://www.someplace.com/somepage_fr.html`

3. The `DefaultLanguage` , such as `en`, defined in the `magnus.conf` file.

   `http://www.someplace.com/en/somepage.html`

   `http://www.someplace.com/somepage_en.html`

4. If none of these are found, the server tries:

   `http://www.someplace.com/somepage.html`

NOTE Keep in mind when naming your localized files that country codes like CH and TW are converted to lower case and dashes (-) are converted to underscores (_).

Using Other Language Settings

The following directives in the `magnus.conf` file specify language defaults:

Table D-1 Language Settings in magnus.conf

Directive	Values	Description
`ClientLanguage`	`en`, `fr`, `de`, `ja`	Specifies the language in which client messages, such as "Not Found" or "Access denied" are to be expressed. This value is used to determine which `ns-httpd.db` database to use for the localized messages.
`DefaultLanguage`	`en`, `fr`, `de`, `ja`	Specifies the language used if a resource cannot be found for the client language.

Search Information

Search capabilities are supported for the following languages:

- English
- German
- French
- Italian
- Spanish
- Swedish
- Dutch
- Japanese

International Search

To view documents in different character set encodings, users must change the character set encoding for their browsers. In addition, since the text search works with one character set encoding at a time, you might receive inaccurate results when using those features. For best results, use one specific character set for all documents when creating search collections from them.

Searching in Japanese

The following information is specific to searching in Japanese.

Query Operators

This release supports the following query operators for Japanese:

Table D-2 Query operators for Japanese

Operator	Japanese Character
`AND`	Yes
`CONTAINS`	No
`ENDS`	Yes
`MATCHES`	Yes
`NEAR`	Yes
`NEAR/N`	Yes
`NOT`	Yes
`OR`	Yes
`PHRASE`	Yes
`STARTS`	Yes
`STEM`	English only
`SUBSTRING`	Yes
`WILDCARD *`	Yes
`WILDCARD ?`	Yes
`WORD`	Yes

Document Formats

This release supports the following document formats for the Japanese language:

- HTML
- ASCII

- NEWS
- MAIL

NOTE The PDF document format is not supported for Japanese.

Searching in Japanese

The following sections give additional information about searching in the Japanese character set.

Document Encodings

This release supports the following document encodings for the Japanese language:

- `euc-jp`
- `Shift_JIS`

NOTE ISO-2022-JP is not supported.

Search Words

This release supports the following search words:

- kanji
- hiragana
- katakana (full-width and half-width)
- ASCII (full-width and half-width)

The search engine translates half-width katakana to full-width katakana, and translates full-width ASCII to half-width ASCII. Users can use full-width and half-width as the same characters.

This release also supports phrase and sentence search.

Servlet Internationalization

When form data is submitted from a browser to the server using `POST`, the browser:

- url-encodes the `POST` data
- Sets the Content-Type to `application/x-www-form-urlencoded`

- Does not send any charset information in the Content-Type header

On the server side, if a servlet tries to access `POST` data using `getParameter` or `getParameterValues`, the servlet container does not have any information about which character encoding to use for `getParameter` strings.

You can configure iPlanet Web Server 6.0 to instruct the servlet container which character encoding to use for interpreting `POST` data strings. To do this, you would specify the character encodingusing the `parameter-encoding` element in `web-apps.xml`:

```
<parameter-encoding enc="value"/>
```

where *value* can be one of the following:

- `auto` (default)
- `none`
- any valid encoding

These values are described below.

auto

`auto` requires the servlet container to look for some hints regarding the character encoding to be used. The hint can be provided using:

- A request attribute with the name: `com.iplanet.server.http.servlet.parameterEncoding`. The value is of type `String`. The request attribute must be set before any calls to `getParameter()` or `getParameterValues()`. Example:

  ```
  request.setAttribute("com.iplanet.server.http.servlet.
  parameterEncoding", "Shift_JIS");
  request.getParameter("test");
  ```

  This option is used if the servlet knows beforehand what the charset of the posted data is.

- A `j_encoding` parameter in the form data. The form that is being submitted can have a hidden element. Example:

  ```
   <input type=hidden name="j_encoding" value="Shift_JIS" >
  ```

This option is typically used if the servlet that is reading the data does not necessarily know what the charset of the posted data is. The hint parameter name, which by default is `j_encoding` can be changed using the`parameter-encoding` element in `web-apps.xml`.

none

Use this option if you wish the platform default encoding to be used for the servlet parameter data.

any valid encoding

If none of the above options are specified, the servlet container interprets this string itself as the encoding, so this can be any valid encoding string like `Shift_JIS`, or `UTF-8`. For example, you would specify this as `UTF-8` if you know that the form `POST` data is always in `UTF-8`.

NOTE The server will always try to resolve the charset from the Content-Type header of the request first.

For more information on `parameter-encoding`, see the *Programmer's Guide to Servlets*.

Posting to JSPs

You can configure `parameter-encoding` to work the same way when you are posting to a JSP instead of a servlet. The following example demonstrates a JSP configuration of ‘`auto`’ to read parameters which are in Japanese `Shift_JIS` encoding:

```
<%@ page contentType="text/html;charset=Shift_JIS" %>
<html>
<head>
<title>JSP Test Case</title>
</head>
<body>
<%
request.setAttribute(“com.iplanet.server.http.servlet.parameterEnco
ding", "Shift_JIS");
```

```
%>
<h1>The Entered Name is : <%= request.getParameter("test") %> </h1>
</body>
</html>
```

Appendix E

Server Extensions for Microsoft FrontPage

This appendix describes using server extensions on your iPlanet Web Server that provide support for Microsoft's FrontPage. These extensions provide the internal server-side support you need if you are using FrontPage webs.

This appendix includes the following sections:

- Overview
- Downloading the Extensions
- Installing FrontPage Server Extensions
- Further Information

Overview

FrontPage server extensions are CGI programs that provide iPlanet Web Server support for FrontPage webs. Client-server communication takes place through standard HTTP POST requests that are forwarded to the appropriate extension's CGI program. If you use FrontPage webs, the extensions provide support for FrontPage authoring and publishing, access permission, and WebBot functions. For example:

- When a user moves a page between folders in a FrontPage web, the extensions automatically update all links to that page from every other page in the web.
- You can specify which users have permission to administer, author or browse a FrontPage web.
- When FrontPage web users participate in a discussion group, the extensions take advantage of the available WebBots to maintain an index of links to discussion articles, tables of contents, and search forms.

The extensions can minimize file transfers over the Internet. For example, when a user opens a FrontPage web from an iPlanet Web Server with the extensions, web metadata, such as its map of links, is downloaded to the user's machine but the full set of web pages remain on the server. A page is downloaded only when it is opened for editing.

Once you have installed the extensions on your server, FrontPage web publishing, administering, and discussion group functionality is available from any computer that is on the Internet or a local Intranet, although you need the FrontPage client program for authoring and administrative functions.

This section includes the following topics:

- Types of FrontPage Webs
- Domain Names and FrontPage Webs
- Security Issues

Types of FrontPage Webs

There are two kinds of FrontPage webs:

- Root webs are the top-level content directory of a Web server or, in a multi-hosting environment, of a virtual Web server. There can only be one root web per Web Server or virtual Web server. A single root web can support a number of sub-webs.
- Sub-webs are complete FrontPage webs that are subdirectories of the root web. Sub-webs can only exist one level below the root web. Each sub-web can have many levels of subdirectories, making up its content.

Even though sub-webs appear below the root web in the Web server's file system and URL space, the root web does not include the content in its sub-webs. This separation of content is done by the FrontPage Server Extensions.

The root web and all sub-webs on a server must have separate copies of the extensions installed or have stub executables of the extensions programs. Having separate copies of the extensions for each FrontPage web lets the server administrator enforce different end-user, author, and administrator permissions on each FrontPage web, since FrontPage uses the server's built-in security mechanism to control access.

Domain Names and FrontPage Webs

FrontPage webs can be implemented on an iPlanet Web Server and accessed by web browsers in the following ways:

- As private domain names, such as `www.mycompany.com`. These are usually implemented as virtual servers on the same physical server machine using multi-hosting. Private domain name customers each get their own root web and have the option of creating sub-webs.
- As a common or shared domain but with private virtual servers, as in `www.mycompany.myprovider.com`, where `myprovider.com` is a shared domain and `www.mycompany` is a private virtual server. Private virtual server customers on a shared domain each get their own root web and have the option of creating sub-webs.
- As a URL on an Internet service provider's server machine, as in `www.myprovider.com/mycompany`. URL customers get a single sub-web.

Security Issues

FrontPage implements web security on your web server by changing the access-control lists (ACLs) for all files and directories in each FrontPage web. Installing FrontPage always modifies the ACLs of the Server Extensions stub executables contained in the `/_vti_bin` directory in each web. A new installation of FrontPage will additionally modify the ACLs of the web content files, but an upgrade of an existing installation of the Server Extensions will not modify the content file ACLs and consequently will leave the security settings at a less secure level than the default FrontPage settings. You can upgrade the ACLs of your web content by using the Check and Fix option of the FrontPage Server Administrator utility.

In addition to modifying the security ACLs of the web content files, FrontPage modifies the ACLs of any system DLLs that are used as a result of a FrontPage DLL call, to ensure that the system DLLs will have the correct level of permissions to run under any administrator, author, or end-user's account. For the complete set of ACLs set on FrontPage files, along with a discussion of security considerations when installing the Server Extensions and the reasons why the ACLs of the system DLLs must be modified, see the additional resources available at Ready-to-Run Software and Microsoft's web sites.

Downloading the Extensions

The first step towards installing the extensions is to download them. You can use Microsoft's FrontPage sites or, if you want to install the Unix /Linux extensions, you can use Ready-to-Run Software's site, which also provides a great deal of information and instruction.

- FrontPage 97 Server Extensions (version 2.0):
    - [NT] You can download an executable file.
    - [Unix /Linux] You can download from Ready-to-Run Software's web site an install script and a set of server extensions. Download two tar files for your platform (for Solaris, they are `vt20.solaris.tar.z` and `wpp.solaris.tar.z`, which is part of the WPP Kit Software).
    - [Unix /Linux] You can download from Microsoft's web site an install script and a set of server extensions. Download two tar files for your platform (for Solaris, they are `vt20.solaris.tar.z` and `wpp.solaris.tar.z`, which is part of the WPP Kit Software.)
- FrontPage 98 Server Extensions (version 3.0):
    - [NT] You can download an executable file.
    - [Unix /Linux] You can download from Ready-to-Run Software's web site an install script and a set of server extensions. Download the `fp_install.sh` file and the tar file for your platform (for Solaris, it is `fp30.solaris.tar.z`)
    - [Unix /Linux] You can download from Microsoft's web site an install script and a set of server extensions. Download the fp_install.sh file and the tar file for your platform (for Solaris, it is fp30.solaris.tar.z).
- FrontPage 2000 Server Extensions (version 4.0):
    - [NT] You can download an executable file, `fp2kserk.exe`, which gives information on how to set up and use a FrontPage-extended web. You can download a set of server extensions from the Microsoft web site, `fpse2k_x86_ENG.exe`.
    - [Unix /Linux] You can download an install script and a set of server extensions from the Ready-to-Run Software web site. Download the `fp_install.sh` file and the tar file for your platform (for Solaris, it is `fp40.solaris.tar.Z`).
    - [Unix /Linux] You can download an install script and a set of server extensions from the Microsoft web site. Download the `fp_install.sh` file and the tar file for your platform (for Solaris, it is `fp40.solaris.tar.Z`).

Before you install the FrontPage Server Extensions, you need to be sure you have enough disk space available on your local machine, that you have a document root directory, that you have enabled authentication, and that you are aware of some important post-install issues such as access permissions.

This section includes the following topics:

- Space Requirements
- Preliminary Tasks
- Some Additional Considerations

Space Requirements

On Windows NT systems, you need to have approximately 6MB of disk space available. The downloaded file is 3MB and the installed files total 2.5MB.

On Unix /Linux systems, you should have at least 32MB available on your server. The Unix /Linux FrontPage extensions need 9MB of disk space in the `/usr/local/frontpage` directory. If you install the extensions onto your web content, you need an extra 5MB per virtual host unless your web content is in the same disk partition as `/usr/local/frontpage`.

Preliminary Tasks

You need to have a document root directory for your iPlanet Web Server, which is created when you start up your server for the first time. This means you must start up your server at least once before installing the extensions.

When the document root exists, replace `$docroot` in `NameTrans fn=document root root=”$docroot”` with the absolute path to the document directory in the `obj.conf` file of the web server configuration directory.

Some Additional Considerations

- Do not remove any of the internal files needed by FrontPage such as the `.nsconfig` file. Doing so disables access control for content upload.
- You cannot set a web to be restricted to valid end-users only. If you set this, you receive a message that says "This server does not support restricting end user access."
- [Unix /Linux only] When you install the stub extensions, you should set the web owner to be the same as the iPlanet Web Server user. This is so that the FrontPage extensions have write permissions to certain directories, namely the `https-`*instance*`/config` directory and the doc root. The `fpsrvadm.exe` script, which installs stub extensions to the webs, asks for the web owner.

Installing FrontPage Server Extensions

You can install the FrontPage 97, the FrontPage 98, or FrontPage 2000 extensions on Windows NT or Unix /Linux platforms. This document provides instructions for the following platforms:

- Windows NT systems
- Unix /Linux systems - FrontPage97 extensions
- Unix /Linux systems - FrontPage98 extensions
- Unix /Linux systems - FrontPage2000 extensions

Installing FrontPage Server Extensions on Windows NT Systems

The installation process for the FrontPage97, FrontPage98, and FrontPage 2000 extensions on a Windows NT system is relatively straightforward. You download and run an executable file, which installs several files and folders on your system. The extensions require a specific directory structure, which is discussed later in this section. After installation, you must perform some additional administrative tasks for setting permissions and accessing specific webs.

These installation instructions are for the standalone FrontPage Server Extensions that are in a self-extracting executable that is downloadable from the Microsoft FrontPage web site.

NOTE You must log into your NT system as Administrator or have administrator permission to install the FrontPage Server Extensions.

Make sure to install one copy of FrontPage and the Server Extension Resource Kit on the same machine as your web server. To install FrontPage Server Extensions on Windows NT, perform the following steps:

1. Run the server extensions setup program for your language and processor type.

   For example, for English FrontPage98 extensions, it is the `fp98ext_x86_enu.exe` file. The server extensions are copied to the folder `C:\Program Files\MicrosoftFrontPage\Version 3.0`. For English FrontPage2000 extensions, it is the `fpse2k_x86_ENG.exe` file. The server extensions are copied to the folder `C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\40`.

2. After installing FrontPage 2000 with the Server Extensions Resource Kit, launch the Server Extensions Administrator under Start | Programs | Administrative Tools, right-click on your machine’s host name under Console Root | FrontPage Server Extensions, and select New Web. Follow the steps in the wizard to select your server instance and configure it for FrontPage Server Extensions.

3. Select the virtual servers on which the FrontPage Server Extensions should be installed and click OK.

4. Enter the name of a new FrontPage administrator account and a password.

You can add other administrator accounts after installing the server extensions using the Permissions command in the FrontPage Explorer.

Installing the server extensions on each FrontPage web may take several minutes and may increase the CPU load on your computer. If this is a new installation of the FrontPage Server Extensions, each page’s contents are parsed to expand FrontPage components (such as Include components and Substitution components), create a hyperlink map of the FrontPage web, and extract page titles and base URLs.

The installation also updates the text indices and recalculates the links in the Web, adds a FrontPage administration account, password, and IP address restriction, and reminds the web administrator to restart the server if new `ObjectType` directives were added to the `obj.conf` file.

For FrontPage97 extensions, these components are installed in the `C:\Program Files\Microsoft FrontPage` directory, where C is your default hard drive. The components are as follows:

- The FrontPage Server Extension `.dll` and `.exe` files are copied to the `\bin` subdirectory and to the default `\windows\system` directory.
- The ISAPI (`.dll`) or CGI (`.exe`) files used by FrontPage to implement the Server Extensions functionality in the user's webs are copied to the `\isapi` and `\_vti_bin` directories, respectively. They are also copied into the document root of each virtual server on which you are installing the FrontPage extensions.
- The FrontPage Server Administrator (`fpsrvwin.exe`) and a command line version (`fpsrvadm.exe`) are copied to the `\bin` directory. The FrontPage Server Administrator is a tool for installing, updating, verifying, or removing the FrontPage Server Extensions.

For FrontPage98 extensions, these components are installed in the `C:\Program Files\Microsoft FrontPage\version 3.0` directory, where C is your default hard drive:

- The FrontPage Server Extensions `.dll` and `.exe` files are copied to the `\bin` subdirectory and to the default `\windows\system` directory.
- The three ISAPI (`.dll`) or CGI (`.exe`) files used by FrontPage to implement the Server Extensions functionality in the user's webs are copied to the `\isapi` and `\_vti_bin` directories, respectively. They are also copied into the document root of each virtual server on which the FrontPage extensions are installed.
- The FrontPage Server Administrator (`fpsrvwin.exe`) and a command line version (`fpsrvadm.exe`) are copied to the `\bin` directory. The FrontPage Server Administrator is a tool for installing, updating, verifying, or removing the FrontPage Server Extensions.
- The Server Extensions Resource Kit.
- HTML Administration forms, a set of HTML forms for remotely administering the FrontPage Server Extensions via web browsers. Also a command line utility (`fpremadm.exe`) for remote administration of the FrontPage Server Extensions is installed in the `\bin` directory.

For FrontPage2000 extensions, these components are installed in the `C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\40` directory, where C is your default hard drive:

- The FrontPage Server Extensions `.dll` and `.exe` files are copied to the `\bin` subdirectory and to the `\WINNT\system32` directory.
- The three ISAPI (`.dll`) or CGI (`.exe`) files used by FrontPage to implement the Server Extensions functionality in the user's webs are copied to the `\isapi` and `\_vti_bin` directories, respectively. They are also copied into the document root of each virtual server on which the FrontPage extensions are installed.
- The FrontPage Server Administrator (`fpsrvwin.exe`) is not used or installed with FrontPage 2000. You must use the Server Extensions Administrator (under Start | Programs | Administrative Tools) or the command line version (`fpsrvadm.exe`), which is copied to the `\bin` subdirectory. The FrontPage Server Administrator is a tool for installing, updating, verifying, or removing the FrontPage Server Extensions.
- The Server Extensions Resource Kit is copied to the `\serk` subdirectory.
- HTML Administration forms, a set of HTML forms for remotely administering the FrontPage Server Extensions via web browsers, are copied to the `\admcgi` and `\admisapi` subdirectories. Also a command line utility (`fpremadm.exe`) for remote administration of the FrontPage Server Extensions is installed in the `\bin` directory.

Installation also modifies or adds the following files and directories:

- Modifies the `magnus.conf` file
- Modifies the server's configuration file (`obj.conf`) to add `ObjectType` directives, marking three of the `\_vti_` directories as containing executables.
- Adds the following seven subdirectories are created under your server's document root:

  ```
  \_private
  \_vti_bin (contains shtml.exe)
  \_vti_bin\_vti_adm (contains admin.exe)
  \_vti_bin\_vti_aut (contains author.exe)
  \_vti_cnf
  \_vti_log
  \_vti_pvt
  \_vti_txt
  \images
  ```

- Creates `.nsconfig` files in the `\_vti_bin`, `\_vti_adm`,`\_vti_aut` directories and the document root directory.

Once you have completed the installation process, you must also perform the following administrative tasks:

- For FrontPage 97 and 98, execute the `fpsrvwin.exe` file (located in the \bin directory of your FrontPage directory) to set the server port, test the extensions, install the extensions to other virtual servers, and update extensions.
- For FrontPage 2000, run the Server Extensions Administrator (under Start | Programs | Administrative Tools) or the command line version (`fpsrvadm.exe`).
- Select the server and web you want to work with:
  - A remote machine must have the FrontPage97, FrontPage98, or FrontPage 2000 program installed (Macintosh or Windows only). Once the FrontPage program is started, the user is prompted for the name of a server to edit or open.
  - If the user wants to edit a web on a different machine, click on “MoreWebs”, on the line to select a web server or disk location enter the *servername:portnumber* of the web to edit then click OK.
  - Select the web you wish to edit from the list of webs on the host machine.
- You need to provision each additional web. You can do this from the client side, with the FrontPage client provided the client has the right authorization (the administrator username and password) for the root web. You can also provision user webs from the server side by using the program `fpsrvadm.exe` to set the password for an individual web. You need to make sure that the new FrontPage web does not inherit the administrator username and password from the root web.
- Locate the `fpadmin.htm` file, typically in the `\admin\cgi` directory (97 and 98) or `\admcgi` directory (2000) of your FrontPage program directory. You can use this to configure your FrontPage web.
- Users can edit the local web that is displayed when FrontPage is started, but they must have a valid user ID and password to modify it.

Installing FrontPage97 Server Extensions on Unix /Linux Systems

The installation process on a Unix /Linux system requires you to have the appropriate file permissions and directories set up beforehand. The extensions require a specific directory structure, which is discussed later in this section. After installation, you must perform some additional administrative tasks for setting permissions and accessing specific webs.

These installation instructions are for the standalone FrontPage Server Extensions that are in a tarred file that is downloadable from the Microsoft FrontPage web site or the Ready-To-Run Software website.

NOTE You must be logged in as the root user to perform the installation. Also, the root user must have write permission for the `/usr/local` directory, even if this is not the directory where you will install the extensions. If you install the extensions in a different directory, a soft link is added automatically to `/usr/local`.

To install the extensions, perform the following steps:

1. Log in as the root user so you can install the FrontPage Server Extensions from the tar file:

   ```
   cd /usr/local
   ```

2. Untar the downloaded file.

   This creates a `/usr/local/FrontPage/version2.0` directory and installs several other new directories under the document root directory. For example, for the FrontPage 97 extensions on a Solaris platform, you untar the vt20.solaris.tar.z file:

   ```
   tar xvf /usr/tmp/vt20.solaris.tar
   ```

3. Change directories to `/usr/local/frontpage/version2.0`.

   ```
   cd frontpage/version2.0
   ```

4. Create a directory named `/extensions` and move the `_vti_bin` directory into it.

   ```
   mv _vti_bin extensions
   ```

5. Install the WPP kit to `/usr/local/frontpage/version2.0`.

   For Solaris, use this code:

   ```
   tar xvf /usr/tmp/wpp.solaris.tar
   ```

6. Rename the directory /executables (`/usr/local/frontpage/version2.0/executables`) to `/_vti_bin`:

   ```
   mv executables _vti_bin
   ```

7. Move the file `fpsrvadm.suid.exe` to the `/bin` directory:

   ```
   mv fpsrvadm.suid.exe bin
   ```

8. Run the `fp_install.sh` shell program and follow the on-screen instructions, which ask for the information described in the following table.

When you are prompted for the name of the server configuration file, enter the pathname of your server’s `magnus.conf` file.

Table E-1 Installation parameter information

`-fpdir <dir>`	default	FrontPage Directory
`-httpdconfdir <dir>`	default	Directory where server’s configuration file is located
`-web <webname>`	required	Web where the Server Extensions are being installed (/ for root web)
`-user <webowner>`	required	User ID of the web owner
`-group <webgroup>`	optional	GroupID of the web owner
`-host <host>`		Name of virtual host where the Server Extensions are being installed. The host specified should be the same as that specified by the Virtual Host directive in the server’s `httpd.conf` file.
`-admuser <fpadmin>`	required	FrontPage Administrator user name
`-admpass <fppass>`	required	FrontPage Administrator password
`-admaddr <ipaddr>`	optional	IP address restriction of FrontPage Administrator. If not IP address mask is specified, the FrontPage Administrator will have access from all IP addresses.

Installing the Server Extensions on each FrontPage web may take several minutes and may increase the CPU load on your computer. If this is a new installation of the FrontPage Server Extensions, each page’s contents are parsed to expand FrontPage components, such as Include components and Substitution components, create a hyperlink map of the FrontPage web, and extract page titles and base URLs.

The installation process also updates the text indices and recalculates the links in the Web, adds a FrontPage administration account, password, and IP address restriction, and reminds the web administrator to restart the server if new `ObjectType` directives were added to the `obj.conf` file.

During installation, the install shell modifies or adds the following files and directories:

- Modifies `magnus.conf`
- Creates a configuration file named `/usr/local/frontpage/`*hostname*`:`*port*`.cnf`
- Modifies the server’s configuration file (`obj.conf`) to add `ObjectType` directives, marking three of the `/_vti_` directories as containing executables.

- Adds seven subdirectories under the server's document root:

  ```
  /_vti_bin (contains shtml.exe)
  /_vti_bin/_vti_adm (contains admin.exe)
  /_vti_bin/_vti_aut (contains author.exe)
  /_vti_cnf
  /_vti_pvt
  /_private
  /_vti_log
  /_vti_txt
  /images
  ```

- Creates `.nsconfig` files in the `/_vti_bin`, `/_vti_adm`, `/_vti_aut` and the document root directories.

Once you have completed the installation process, you must perform the following administrative tasks:

1. Execute the `fpsrvwin.exe` file to set the server port, test the extensions, install the extensions to other virtual servers, and update extensions.
2. A remote machine must have the FrontPage 97, 98, or 2000 program installed (Macintosh or Windows only).

   Once the FrontPage program is started, the user is prompted for the name of a server to edit or open.
3. If the user wants to edit a web on a different machine, click on "MoreWebs" on the line to select a web server or disk location enter in the *servername:portnumber* of the web to edit. Choose OK.
4. Select the proper web from the list of webs on the host machine to edit.

Installing FrontPage98 Server Extensions on Unix /Linux Systems

These installation instructions are for the stand-alone FrontPage Server Extensions that are in a tarred file that is downloadable from the Microsoft FrontPage web site or the Ready-To-Run Software website.

After installation, you must perform some additional administrative tasks for setting permissions and accessing specific webs.

NOTE You must be logged in as the root user to perform the installation. Also, the root user must have write permission for the `/usr/local` directory, even if this is not the directory where you will install the extensions. If you install the extensions in a different directory, a soft link is added automatically to `/usr/local`.

To install the extensions, perform the following steps:

1. Log in as the root user so you can install the FrontPage Server Extensions from the tar file.

2. Enter `cd /usr/local`, or `cd` to the directory where the two downloaded files (`fp30.solaris.tar.Z` and `fp_install.sh`) are located.

3. Run the `fp_install.sh` shell program and follow the on-screen instructions, which ask for parameter information.

   When you are prompted for the name of the server configuration file, enter the pathname of your server's `magnus.conf` file.

Installing the Server Extensions on each FrontPage web may take several minutes and may increase the CPU load on your computer. If this is a new installation of the FrontPage Server Extensions, each page's contents are parsed to expand FrontPage components, such as Include components and Substitution components, create a hyperlink map of the FrontPage web, and extract page titles and base URLs.

The installation process also updates the text indices and recalculate the links in the Web, adds a FrontPage administration account, password, and IP address restriction, and reminds the web administrator to restart the server if new `ObjectType` directives were added to the `obj.conf` file.

During installation, the install shell modifies or adds the following files and directories:

- Modifies `magnus.conf`
- Creates a configuration file named `/usr/local/frontpage/`*hostname*`:`*port*`.cnf`

- Modifies the server's configuration file (`obj.conf`)to add `ObjectType` directives, marking three of the `/_vti_` directories as containing executables.
- Adds seven subdirectories under the server's document root:

```
/_vti_bin (contains shtml.exe)
/_vti_bin/_vti_adm (contains admin.exe)
/_vti_bin/_vti_aut (contains author.exe)
/_vti_cnf
/_vti_pvt
/_private
/_vti_log
/_vti_txt
/images
```

Creates .`nsconfig` files in the `/_vti_bin`, `/_vti_adm`, `/_vti_aut` and the document root directories.

Installing FrontPage2000 Server Extensions on Unix /Linux Systems

These installation instructions are for the stand-alone FrontPage Server Extensions that are in a tarred file that is downloadable from the Microsoft FrontPage web site or the Ready-To-Run Software website.

After installation, you must perform some additional administrative tasks for setting permissions and accessing specific webs.

NOTE You must be logged in as the root user to perform the installation. Also, the root user must have write permission for the `/usr/local` directory, even if this is not the directory where you will install the extensions. If you install the extensions in a different directory, a soft link is added automatically to `/usr/local`.

To install the extensions, perform the following steps:

1. Log in as the root user so you can install the FrontPage Server Extensions from the tar file.
2. Enter `cd /usr/local`, or `cd` to the directory where the two downloaded files (`fp40.solaris.tar.Z` and `fp_install.sh`) are located.

3. Run the `fp_install.sh` shell program and follow the on-screen instructions, which ask for parameter information.

4. Enter the pathname of your server's `magnus.conf` file when you are prompted for the name of the server configuration file.

Installing the Server Extensions on each FrontPage web may take several minutes and may increase the CPU load on your computer. If this is a new installation of the FrontPage Server Extensions, each page's contents are parsed to expand FrontPage components, such as Include components and Substitution components, create a hyperlink map of the FrontPage web, and extract page titles and base URLs.

The installation process also updates the text indices and recalculate the links in the Web, adds a FrontPage administration account, password, and IP address restriction, and reminds the web administrator to restart the server if new `ObjectType` directives were added to the `obj.conf` file.

During installation, the install shell modifies or adds the following files and directories:

- Modifies `magnus.conf`
- Creates a configuration file named `/usr/local/frontpage/`*hostname*`:`*port*`.cnf`
- Modifies the server's configuration file (`obj.conf`)to add `ObjectType` directives, marking three of the `/_vti_` directories as containing executables.
- Adds seven subdirectories under the server's document root:

```
/_vti_bin (contains shtml.exe)
/_vti_bin/_vti_adm (contains admin.exe)
/_vti_bin/_vti_aut (contains author.exe)
/_vti_cnf
/_vti_pvt
/_private
/_vti_log
/_vti_txt
/images
```

Creates .`nsconfig` files in the `/_vti_bin`, `/_vti_adm`, `/_vti_aut` and the document root directories.

Further Information

Additional detailed information can be obtained from Microsoft’s FrontPage web site:

```
http://www.microsoft.com
```

For Unix/ Linux only, information can also be obtained from the Ready-to-Run Software web site:

```
http://www.rtr.com
```

Glossary

Access Control Entries (ACEs) A hierarchy of rules which the web server uses to evaluate incoming access requests.

Access Control List (ACL) A collection of ACEs. An ACL is a mechanism for defining which users have access to your server. You can define ACL rules that are specific to a particular file or directory, granting or denying access to one or more users and groups.

admpw The username and password file for the Enterprise Administrator Server superuser.

agent Software that runs the network-management software in a network device, such as a router, host, or X terminal. See also intelligent agents.

authentication Allows clients to verify their identity to the server. Basic or Default authentication requires users to enter a username and password to access your web server or web site. It requires a list of users and groups in an LDAP database. See also digest and SSL authentication.

The granting of access to an entire server or particular files and directories on it. Authorization can be restricted by criteria including hostnames and IP addresses.

browser See client.

cache A copy of original data that is stored locally. Cached data doesn't have to be retrieved from a remote server again when requested.

certificate A nontransferable, nonforgeable, digital file issued from a third party that both communicating parties already trust.

certification authority (CA) An internal or third-party organization that issues digital files used for encrypted transactions.

Certificate revocation list (CRL) CA list, provided by the CA, of all revoked certificates.

Compromised key list (CKL) A list of key information about users who have compromised keys. The CA also provides this list.

CGI Common Gateway Interface. An interface by which external programs communicate with the HTTP server. Programs that are written to use CGI are called CGI programs or CGI scripts. CGI programs handle forms or parse output the server does not normally handle or parse.

chroot An additional root directory you can create to limit the server to specific directories. You'd use this feature to safeguard an unprotected server.

cipher A cipher is a cryptographic algorithm (a mathematical function), used for encryption or decryption.

ciphertext Information disguised by encryption, which only the intended recipient can decrypt.

client Software, such as Netscape Navigator, used to request and view World Wide Web material. Also known as a browser program.

client auth Client authentication.

cluster A group of remote 'slave' administration servers added to and controlled by a 'master' andministration server. All servers in a cluster must be of the same platform and have the same userid and password.

collection A database that contains information about documents, such as word list and file properties. Collections are used by the search function to retrieve documents matching specified search criteria.

Common LogFile Format The format used by the server for entering information into the access logs. The format is the same among all major servers, including the iPlanet Web Server.

DHCP Dynamic Host Configuration Protocol. An Internet Proposed Standard Protocol that allows a system to dynamically assign an IP address to individual computers on a network.

daemon (Unix) A background process responsible for a particular system task.

digest authentication. Allows the user to authenticate without sending the username and password as cleartext. The browser uses the MD5 algorithm to create a digest value. The server uses the Digest Authentication plug-in to compare the digest value provided by the client.

DNS Domain Name System. The system that machines on a network use to associate standard IP addresses (such as 198.93.93.10) with hostnames (such as `www.iplanet.com`). Machines normally get this translated information from a DNS server, or they look it up in tables maintained on their systems.

DNS alias A hostname that the DNS server knows points to a different host—specifically a DNS CNAME record. Machines always have one real name, but they can have one or more aliases. For example, an alias such as `www.`*yourdomain.domain* might point to a real machine called `realthing.`*yourdomain.domain* where the server currently exists.

document root A directory on the server machine that contains the files, images, and data you want to present to users accessing the server.

drop word See stop word.

encryption The process of transforming information so it can't be decrypted or read by anyone but the intended recipient.

Administration Server A web-based server that contains the forms you use to configure all of your iPlanet Web Servers.

expires header The expiration time of the returned document, specified by the remote server.

extranet An extension of a company's intranet onto the Internet, to allow customers, suppliers, and remote workers access to the data.

file extension The last part of a filename that typically defines the type of file. For example, in the filename `index.html` the file extension is `html`.

file type The format of a given file. For example, a graphics file doesn't have the same file type as a text file. File types are usually identified by the file extension (`.gif` or `.html`).

firewall A network configuration, usually both hardware and software, that protects networked computers within an organization from outside access. Firewalls are commonly used to protect information such as a network's email and data files within a physical building or organization site.

flexible log format A format used by the server for entering information into the access logs.

FORTEZZA An encryption system used by U.S. government agencies to manage sensitive but unclassified information.

FTP File Transfer Protocol. An Internet protocol that allows files to be transferred from one computer to another over a network.

GIF Graphics Interchange Format. A cross-platform image format originally created by CompuServe. GIF files are usually much smaller in size than other graphic file types (BMP, TIFF). GIF is one of the most common interchange formats. GIF images are readily viewable on Unix, Microsoft Windows, and Apple Macintosh systems.

hard restart The termination of a process or service and its subsequent restart. See also soft restart.

home page A document that exists on the server and acts as a catalog or entry point for the server's contents. The location of this document is defined within the server's configuration files.

hostname A name for a machine in the form *machine.domain.dom*, which is translated into an IP address. For example, `www.iplanet.com` is the machine `www` in the subdomain `iplanet` and `com` domain.

HTML Hypertext Markup Language. A formatting language used for documents on the World Wide Web. HTML files are plain text files with formatting codes that tell browsers such as the Netscape Navigator how to display text, position graphics and form items, and display links to other pages.

HTTP HyperText Transfer Protocol. The method for exchanging information between HTTP servers and clients.

HTTP-NG The next generation of HyperText Transfer Protocol.

HTTPD An abbreviation for the HTTP daemon or service, a program that serves information using the HTTP protocol. The iPlanet Web Server is often called an HTTPD.

HTTPS A secure version of HTTP, implemented using the Secure Sockets Layer, SSL.

imagemap A process that makes areas of an image active, letting users navigate and obtain information by clicking the different regions of the image with a mouse. Imagemap can also refer to a CGI program called "imagemap," which is used to handle imagemap functionality in other HTTPD implementations.

inittab (Unix) A Unix file listing programs that need to be restarted if they stop for any reason It ensures that a program runs continuously. Because of its location, it is also called `/etc/inittab`. This file isn't available on all Unix systems.

intelligent agent An object within a server that performs various requests (such as HTTP, NNTP, SMTP, and FTP requests) on behalf of the user. In a sense, the intelligent agent acts as a client to the server, making requests that the server fulfills.

IP address Internet Protocol address. A set of numbers, separated by dots, that specifies the actual location of a machine on the Internet (for example, 198.93.93.10).

ISDN Integrated Services Digital Network.

ISINDEX An HTML tag that turns on searching in the client. Documents can use a network navigator's capabilities to accept a search string and send it to the server to access a searchable index without using forms. In order to use <ISINDEX>, you must create a query handler.

ISMAP ISMAP is an extension to the IMG SRC tag used in an HTML document to tell the server that the named image is an imagemap.

ISP Internet Service Provider. An organization that provides Internet connectivity.

Java An object-oriented programming language created by Sun Microsystems used to create real-time, interactive programs called applets.

JavaScript A compact, object-based scripting language for developing client and server Internet applications.

JavaServer Pages Extensions that enable all JavaServer page metafunctions, including instantiation, initialization, destruction, access from other components, and configuration management. JavaServer pages, are reusable Java applications that run on a web server rather than in a web browser.

Java Servlets Extensions that enable all Java servlet metafunctions, including instantiation, initialization, destruction, access from other components, and configuration management. Java servlets are reusable Java applications that run on a web server rather than in a web browser.

last-modified header The last modification time of the document file, returned in the HTTP response from the server.

LDAP database A database where lists of users and gorups is stored for use in authentication.

listen socket The combination of port number and IP address. Connections between the server and clients happen on a listen socket.

magnus.conf The main Enterprise Server configuration file. This file contains global server configuration information (such as, port, security, and so on). This file sets the values for variables that configure the server during initialization. Enterprise Sever reads this file and executes the variable settings on startup. The server does not read this file again until it is restarted, so you must restart the server every time you make changes to this file.

MD5 A message digest algorithm by RSA Data Security. MD5 can be used to produce a short digest of data that is unique with high probability. It is mathematically extremely hard to produce a piece of data that produces the same message digest email.

MD5 signature A message digest produced by the MD5 algorithm.

MIB Management Information Base.

MIME Multi-Purpose Internet Mail Extensions. An emerging standard for multimedia email and messaging.

mime.types The MIME (Multi-purpose Internet Mail Extension) type configuration file. This file maps file extensions to MIME types, to enable the server to determine the type of content being requested. For example, requests for resources with `.html` extensions indicate that the client is requesting an HTML file, while requests for resources with `.gif` extensions indicate that the client is requesting an image file in GIF format.

modutil Software utility required for installing PKCS#11 module for external encryption or hardware accelerator devices.

MTA Message Transfer Agent. You must define your server's MTA Host to use agent services on your server.

Netscape Console A Java application that provides server administrators with a graphical interface for managing all Netscape servers from one central location anywhere within your enterprise network. From any installed instance of Netscape Console, you can see and access all the Netscape servers on your enterprise's network to which you have been granted access rights.

NIS (Unix) Network Information Service. A system of programs and data files that Unix machines use to collect, collate, and share specific information about machines, users, file systems, and network parameters throughout a network of computers.

network management station (NMS) A machine users can use to remotely manage a network. A managed device is anything that runs SNMP such as hosts, routers, and iPlanet servers. An NMS is usually a powerful workstation with one or more network management applications installed.

NNTP Network News Transfer Protocol for newsgroups. You must define your news server host to use agent services on your server.

NSAPI See Server Plug-in API.

obj.conf The server's object configuration file. This file contains additional initialization information, settings for server customization, and instructions that the server uses to process requests from clients (such as browsers). iPlanet Web Server reads this file every time it processes a client request.

password file (Unix) A file on Unix machines that stores Unix user login names, passwords, and user ID numbers. It is also known as `/etc/passwd`, because of where it is kept.

pk12util Software utility required to export the certificate and key databases from your internal machine, and import them into an external PKCS#11 module.

primary document directory See document root.

protocol A set of rules that describes how devices on a network exchange information.

private key The decryption key used in public-key encryption.

public key The encryption key used in public-key encryption.

public information directories (Unix) Directories not inside the document root that are in a Unix user's home directory, or directories that are under the user's control.

Quality of Service the performance limits you set for a server instance, virtual server class, or virtual server.

RAM Random access memory. The physical semiconductor-based memory in a computer.

rc.2.d (Unix) A file on Unix machines that describes programs that are run when the machine starts. This file is also called `/etc/rc.2.d` because of its location.

redirection A system by which clients accessing a particular URL are sent to a different location, either on the same server or on a different server. This system is useful if a resource has moved and you want the clients to use the new location transparently. It's also used to maintain the integrity of relative links when directories are accessed without a trailing slash.

resource Any document (URL), directory, or program that the server can access and send to a client that requests it.

RFC Request For Comments. Usually, procedures or standards documents submitted to the Internet community. People can send comments on the technologies before they become accepted standards.

root (Unix) The most privileged user on Unix machines. The root user has complete access privileges to all files on the machine.

server daemon A process that, once running, listens for and accepts requests from clients.

Server Plug-in API An extension that allows you to extend and / or customize the core functionality of iPlanet servers and provide a scalable, efficient mechanism for building interfaces between the HTTP server and back-end applications. Also known as NSAPI.

server root A directory on the server machine dedicated to holding the server program, configuration, maintenance, and information files.

SOCKS Firewall software that establishes a connection from inside a firewall to the outside when direct connection would otherwise be prevented by the firewall software or hardware (for example, the router configuration).

soft restart A way to restart the server that causes the server to internally restart, that is, reread its configuration files. A soft restart sends the process the `HUP` signal (signal number one). The process itself does not die, as it does in a hard restart.

SSL Secure Sockets Layer. A software library establishing a secure connection between two parties (client and server) used to implement HTTPS, the secure version of HTTP.

SSL authentication Confirms users' identities with security certificates by using the information in the client certificate as proof of identity, or verifying a client certificate published in an LDAP directory.

stop word A word identified to the search function as a word not to search on. This typically includes such words as *the, a, an, and.* Also referred to as *drop words.*

strftime A function that converts a date and a time to a string. It's used by the server when appending trailers. `strftime` has a special format language for the date and time that the server can use in a trailer to illustrate a file's last-modified date.

superuser (Unix) The most privileged user available on Unix machines (also called root). The superuser has complete access privileges to all files on the machine.

Sym-links (Unix) Abbreviation for symbolic links, which is a type of redirection used by the Unix operating system. Sym-links let you create a pointer from one part of your file system to an existing file or directory on another part of the file system.

TCP/IP Transmission Control Protocol / Internet Protocol. The main network protocol for the Internet and for enterprise (company) networks.

telnet A protocol where two machines on the network are connected to each other and support terminal emulation for remote login.

timeout A specified time after which the server should give up trying to finish a service routine that appears hung.

TLS Secure Sockets Layer. A software library establishing a secure connection between two parties (client and server) used to implement HTTPS, the secure version of HTTP.

top (Unix) A program on some Unix systems that shows the current state of system resource usage.

top-level domain authority The highest category of hostname classification, usually signifying either the type of organization the domain is (for example, `.com` is a company, `.edu` is an educational institution) or the country of its origin (for example, `.us` is the United States, `.jp` is Japan, `.au` is Australia, `.fi` is Finland).

uid (Unix) A unique number associated with each user on a Unix system.

URI Uniform Resource Identifier. A file identifier that provides an additional layer of security by using an abbreviated URL. The first part of the URL is substituted with a URL mapping that hides the file's full physical pathname from the user. See also URL mapping.

URL Uniform Resource Locator. The addressing system used by the server and the client to request documents. A URL is often called a location. The format of a URL is *protocol://machine:port/document*.

A sample URL is `http://www.iplanet.com/index.html`.

URL database repair A process that repairs and updates a URL database that has been damaged by a software failure, a system crash, a disk breakdown, or a full file system.

URL mapping The process of mapping a document directory's physical pathname to a user-defined alias so that files within the directory need only refer to the directory's alias instead of the file's full physical pathname. Thus, instead of identifying a file as `usr/iplanet/servers/docs/index.html`, you could identify the file as `/myDocs/index.html`. This provides additional security for a server by eliminating the need for users to know the physical location of server files.

virtual server class A collection of virtual servers that shares the same basic configuration information in a obj.conf file.

virtual server Virtual servers are a way of setting up multiple domain names, IP addresses, and server monitoring capabilities with a single installed server.

web application A collection of servlets, JavaServer Pages, HTML documents, and other web resources which might include image files, compressed archives, and other data. A web application may be packaged into an archive (a WAR file) or exist in an open directory structure.

Web Application Archive (WAR) An archive file that contains a complete web application in compressed form. iPlanet Web Server cannot access an application in a WAR file. You must uncompress a web application (deploy it using the `wdeploy` utility) before iPlanet Web Server can serve it.

Windows CGI (Windows NT) CGI programs written in a Windows-based programming language such as Visual Basic.

Index